Furtwangen University

Bachelor Thesis in the field of Online Media

Implementation and evaluation of Neural Radiance Fields as an immersive technology in industry at technology start-up Dromni: An application study

supervised by:	Prof. Dr. Uwe Hahne
cosupervised by:	Dr. Maximilian Prexl
submitted on:	30.08.2024
submitted by:	Dominik Widmann 269961 Moltkestraße 80 77654 Offenburg do.widmann@gmail.com

Abstract

This thesis explores the implementation and evaluation of Neural Radiance Fields (NeRFs) as an immersive technology within industrial applications. The research is centered around the use of the Nerfstudio framework, which consolidates various state-of-the-art NeRF techniques under a unified structure. The work delves into the theoretical foundations of NeRFs, examining improvements and models that enhance the algorithm's ability to synthesize novel views of complex scenes with high precision. A significant part of the paper is dedicated to providing a practical guide for users, especially those without a technical background, on how to effectively use Smartphone cameras for data collection. This includes detailed instructions on setting up the framework, overcoming common installation challenges, and optimizing data acquisition using consumer electronics. The research emphasizes the importance of reproducibility and the impact of various factors on the quality of datasets used in NeRF training. In the applied section, the thesis explores the potential of NeRF technology in industrial contexts through case studies, demonstrating its ability to create immersive 3D representations of environments such as construction sites.

Contents

1 Introduction

In this thesis, we investigate the ability of the Neural Radiance Fields (NeRF) algorithm to convert two-dimensional (2D) images into three-dimensional (3D) scenes and explore versatile ways of creating and interacting with these scenes. As part of this research, we also examine various potential use cases of 3D visualisation in a range of industrial environments. To assess the usability of data captured by smartphone cameras (SPCs) in such settings, we generate datasets under realistic conditions and analyze their completeness and level of detail. This process involves recording both photos and videos, capturing specific key objects, and conducting comprehensive spatial recordings. Additionally, we compare factors critical for the coverage and density of the subsequent 3D reconstruction, including the number of images, capture speed, and elements or areas within the scene that may not be visible or detectable by the camera.

1.1 Context and Technological Evolution

The Industrial Revolution of the 18th and 19th centuries marked the beginning of an unprecedented phase of technological advancements. The mechanization of production facilities led to a massive increase in industrial capacities and laid the foundation for a series of subsequent inventions. Within a century, innovations such as the electromagnetic telegraph (Samuel Morse, 1837), the internal combustion engine (Henry Ford, 1888), and the first transatlantic commercial flight (1939) revolutionized communication, travel, and trade. These advancements became symbols of progress, dramatically changing the methods and speed with which information, goods, and services were exchanged.

This development was further amplified by the advent of computers in the mid-20th century, representing a new era of technological advancement. The introduction of the internet in the 1990s intensified this transformation even more, connecting the world in ways never seen before. In light of these significant technological shifts, the current developments in Artificial Intelligence (AI), particularly in Large Language Models (LLMs) such as GPT-4, underscore the relevance of this topic. As we explore the potential of AI in various fields, it is clear that it will establish itself in many areas of human activity and communication. This potential is particularly notable in industrial applications, where AI-driven innovations are poised to make a substantial impact [BM14]. This aspect must be considered within the context of human development.

Humans take pride in many things, from overcoming challenging tasks to creating music and art. This is made possible by something that distinguishes us from many other living beings: intelligence. Intelligence is often understood as a trait, like height or endurance, but defining it is usually more complicated. Generally, intelligence is a mechanism that allows us to solve problems—more specifically, to survive challenges such as finding food, securing shelter, competing with others, or fleeing from enemies [Got97].

Intelligence is not a single ability but the interplay of acquiring knowledge, learning new things, being creative, developing strategies, and thinking critically. It manifests in various behaviors, from innate, instinctive reactions to diverse learning methods to self-reflection. However, science is not entirely in agreement on where intelligence begins and what should be considered intelligence [Got97].

Although learning and intelligence can be conceptually distinguished in terms of formal definitions and measurements, a review of evidence on the relationship between individual differences in measures of learning and intelligence suggests that no clear distinction can be made between the cognitive processes contributing to individual differences in these two realms [Jen89]. Intelligence, therefore, remains a complex and multi-faceted concept.

Consider the human brain and intelligence as functioning more like a sandbox—a malleable set of diverse abilities, akin to a virtual machine that is isolated from the rest of the system, designed to execute potentially unsafe ideas without causing any damage to the wider system. The most basic tools in this "sandbox" are the abilities to acquire knowledge, store it, and learn new things from it [Pro24].

We perceive information about the world around us through our senses: we see, hear, smell, feel, and taste. This sensory input is how we navigate the world and respond appropriately to it. Sensory stimuli are the basis for all our actions and behaviors. However, information encompasses much more, provided we can retain and store it. The ability to store and recall information is crucial for efficient learning, allowing previously learned material to be retrieved as needed, thereby accelerating the learning process [Ame21].

The groundbreaking successes achieved through such systems are exemplified by the human immune system, which functions in a manner almost identical to a well-designed learning system. The human immune system can recognize pathogens, remember them, and respond more quickly and effectively upon subsequent encounters. This ability has enabled us to conquer every climate zone, every continent on Earth, and even outer space.

We can remember events, places, and associations, as well as behaviors, such as hunting and gathering. Some of these can only be mastered through countless repetitions. This is learning—a process in which a sequence of thoughts or actions is assembled, essentially a series of repeatable behaviors that can be modified and adapted.

In a remarkably short period of time, our technological advancement has empowered us to overcome boundaries, mountains, lakes, continents, oceans and space. Furthermore, we now possess the ability to access and retain a immeasurable reservoir of knowledge, collectively accumulated by our species, at our fingertips. No other living being on this planet is capable of combining various skills and using them as tools in the way humans do. For the first time in history, we can solve complex tasks on a large scale through the use of AI.

For example, in Germany, around 95.5% of people aged 20–29 own a smartphone. These devices enable the externalization of cognitive processes, a phenomenon known as cognitive offloading [RG16]. Furthermore, it facilitates the externalization of even the most intricate aspects of social interaction, the processing and solving of tasks, and a vast array of other activities. The success of AI-driven tools like ChatGPT, an AI chatbot based on LLMs introduced by OpenAI in November 2022, demonstrates the tip of the iceberg in terms of the development and the power of this externalization. ChatGPT reached an estimated 100 million monthly active users just two months after its launch, making it the fastest-growing consumer application in history [Rao23].

This rapid adoption of AI-driven tools is part of a broader wave of innovations which currently reshaping various industries. In the field of computer graphics, the emergence of 3D scene reconstruction and rendering techniques like NeRF and Gaussian Splatting represents state-of-the-art (SOTA) solutions for addressing the challenges of inverse rendering and novel view synthesis. Meanwhile traditional methods like photogrammetry focus primarily on reconstructing the geometric attributes of objects, often overlooking the intricate details of lighting and color information inherent in radiance fields. NeRFs, by contrast, capture and utilize these radiance fields, enabling more accurate and visually compelling reconstructions from sparse input data and therefore represent a versatile approach to the industrial utilization and creation of 3D representations [Mil+20]. At this point, however, it is clear that AI-supported applications will fundamentally transform the way 3D scenes are created, edited and used. And therefore it is our task to address this development in the sense of a tool in whose development, use and dissemination we can play a part.

1.2 Basics of 3D Visualisation

As a result of technological developments, industry and society are facing an ever-increasing digitalisation. This development, known as Industry 4.0, is rapidly changing the world of manufacturing through the integration of cloud computing, machine learning (ML), artificial intelligence (AI) and universal network connectivity. One of the numerous innovations that have emerged from this development is digital twins (DT). DTs represent a transformative technology that employs software systems to replicate the behaviour of physical processes within a digital environment [RSK20]. The rise of technologies like DTs has driven demand for highquality 3D graphics. While industrial 3D applications (CAD, FEA, scanning, robotics) are valuable, they face high costs and dependence on manual input. However, Industry 4.0 is also associated with other concepts and technologies, including Internet of Things (IoT), additive manufacturing and augmented reality (AR).

Still, these technologies have one thing in common: they either require 3D input data, which has heretofore been created using traditional methods (3D scanning and reconstruction), or they can be used as sensors to generate 3D data.

One of several methods for visualizing virtual objects is three-dimensional (3D) modeling. It is the process of creating digital representations of objects in a three-dimensional space. This process can simulate real-world objects or imaginative designs. 3D models can be created using various techniques, including primitive shapes, modifiers, polygonal modeling, and sculpting. The representation is usually achieved through mathematical manipulation of points in a virtual space. These points, technically known as vertices, form a collection known as a mesh. Polygon (Triangular) Meshes is a collection of faces, vertices, and edges leveraged to define the shape of the polyhedra object [Ros05]. This mesh serves as the foundational framework of the 3D model, determining its overall shape and configuration. The more detailed the mesh, the more intricate and realistic the final 3D model becomes.

At this point in time, however, the model do not include any texture. Texturing involves applying a 'skin' over the mesh of the 3D model. This texture can include details like color, shininess, roughness, and transparency and is therefore responsible for creating the illusion of three-dimensionality, thereby rendering the model more realistic [Ado]. Although 3D modelling has established itself primarily in the field of computer games, the creation of a D3 model usually involves considerable manual effort. So-called game asset libraries such as the Unity Asset Store can simplify this considerably, though they are not able to reproduce existing scenes to the same level of detail. In addition, 3D objects require manual modelling and texturing. The same applies to the creation of realistic lighting and material properties.

Another approach for visualizing objects is Photogrammetry. Photogrammetry represents a wholly different approach to the creation of three-dimensional models, utilising a camera. By capturing an object from multiple angles under optimal lighting conditions and then processing these images through a specialized program, a software can extract feature-points, assign them coordinates, and connect them into a textureless composite mesh of triangular polygons.

To create texture, it is derived from the original two-dimensional images and applied to the mesh as a photographic surface. Together, these elements generate a three-dimensional representation of the photographed object, with data obtained directly from the real world [Art]. However, photogrammetry presents a number of challenges in the field, the most notable being the handling of transparent or reflective objects. Capturing these materials can lead to inaccurate reconstructions due to their unique light interaction properties. It is within this domain that the NeRF algorithm has made a remarkable impact [Mil+20]. Nevertheless, the creation of NeRF remains a sparsely described process that necessitates expertise across a diverse range of disciplines and the utilisation of specialised tools. As a consequence, they remain inaccessible to the general public. In order to overcome this challenge, the creation of datasets suitable for use with NeRF is covered in detail in chapter 4 on *"Smartphone-Based Dataset Acquisition"*. As it is a relatively new technique, NeRF capabilities and limitations are still being investigated.

2 Neural Radiance Fields for View Synthesis

NeRF is a method for synthesizing novel views of complex scenes by optimizing an underlying continuous volumetric scene function using a limited set of 2D images. Given a set of images capturing a static scene from multiple angles, along with their corresponding poses, the neural network learns to represent that scene in a way that allows new views to be synthesized. This means it can produce images from viewpoints and angles that were not part of the original dataset [Mil+20].

The NeRF algorithm represents a continuous scene as a 5D vector-valued function whose input is a 3D spatial location $\mathbf{X} = (x, y, z)$ and a 2D viewing direction $\mathbf{d} = (\phi, \theta)$, and whose output is the emitted color $\mathbf{c} = (r, g, b)$ depending on each direction at each point and a view-independent volume density σ that ranges from $[0, \infty]$.

To approximate this continuous 5D scene representation, a Multilayer Perceptron (MLP) network F^θ is utilized, where $F^\theta : (\mathbf{x}, \mathbf{d}) \rightarrow (\mathbf{c}, \sigma)$. The weights of the network are optimized to map each 5D input coordinate to its corresponding volume density and directional emitted color. Part a and b of Figure 3 provide an overview of the NeRF scene representation [Mil+20].

The following steps are performed: First, 5D coordinates are sampled along the camera rays, and an MLP computes both color and volume density (steps a and b). Next, the output from the MLP is utilized to generate the individual pixels of an image using classical volume rendering techniques (step c). Finally, the full differentiability of the rendering function is utilized to optimize the weights of the MLP (step d). This is achieved by minimizing the loss function, which is simply defined as the total squared error between the rendered and true pixel colors for both the less detailed version and fine rendering:

$$\mathcal{L} = \sum_{r \in R} \left(\hat{C}_c(r) - C_{gt}(r) \right)^2 + \left(\hat{C}_f(r) - C_{gt}(r) \right)^2$$

Where R represents the set of rays in each batch and $C_{gt}(r)$, $\hat{C}_c(r)$, and $\hat{C}_f(r)$ denote the ground-truth, coarse volume predicted, and fine volume predicted pixel colors for ray r, respectively. The ray r passing through a pixel is determined based on the pixel's position and the camera's location. Due to the differentiability of the rendering function, the only input required to optimize this scene representation is a set of images with known camera poses [Mil+20].

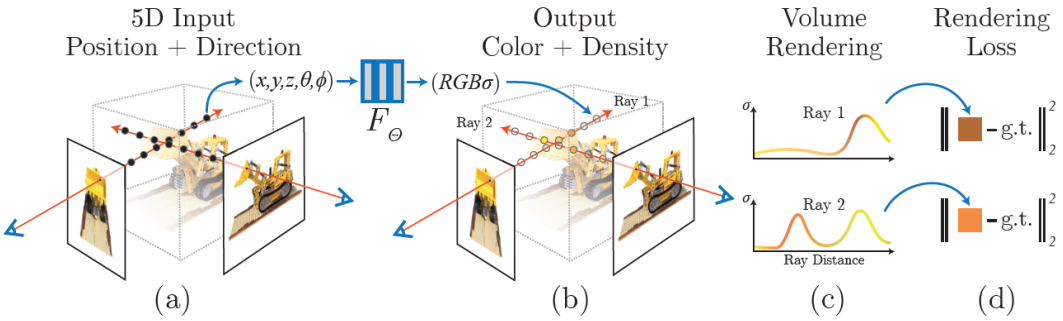


Figure 1: An overview of the neural radiance field scene representation and differentiable rendering procedure. The synthesis of images is accomplished by sampling 5D coordinates, encompassing location and viewing direction, along camera rays (a). These sampled locations are then fed into a MLP to generate color and volume density values (b). By employing volume rendering techniques, these values are composited to create the final image (c). The rendering function is differentiable, enabling the optimization of the scene representation by minimizing the residual between the synthesized images and the observed ground truth images (d) [Mil+20].

Image source: [Mil+20]

2.1 NeRF improvements

Since the introduction of NeRF in 2020 [Mil+20], the model has undergone numerous enhancements and extensions tailored to different applications. These enhancements were specifically designed to address the challenges faced in the early stages of NeRF's development. Key enhancements include methods to increase training efficiency, reduce rendering times, and improve the model's ability to accurately process real-world scenes were made.

2.2 NeRF Models Overview

The generation of realistic 3D scenes from multi-view videos represents a significant challenge for Neural View Synthesis (NVS), particularly in diverse real-world scenarios. Therefore a significant number of recently developed works and approaches are specifically designed to address these limitations of NVS when faced with variations in lighting, reflections, transparency, and overall scene complexity. It is noteworthy that recent research, including the presentation of a comprehensive scene dataset for deep learning-based 3D vision, has furnished invaluable benchmarks for evaluating the advancement of NVS methodologies [Lin23]. The following section examines both, pioneering models, as well as models employed in the aforementioned work.

Original NeRF: The original NeRF was introduced in March 2020 by Ben Mildenhall and his colleagues. It uses a fully connected deep neural network to model a continuous 3D scene. The network takes as input a 3D point and a viewing direction, and outputs a color (RGB) and density value for that point. This allows NeRF to render novel views of complex 3D scenes from 2D images by synthesizing the scene from different perspectives [Mil+20].

NeRF-W: NeRF in the Wild is an extension of the original NeRF model to handle more complex, real-world scenes with uncontrolled illumination. NeRF-W introduces additional input features to represent the spatially-varying lighting conditions, improving the model's ability to generalize across diverse scenes [Mar+20].

NeRF++: The NeRF++ model builds upon the original NeRF by incorporating additional features such as spatially-varying reflectance and incorporating auxiliary tasks to improve training. The main goal of NeRF++ is to enhance the quality of the generated images while maintaining robustness and efficiency [Zha+20].

D-NeRF: Neural Radiance Fields for Dynamic Scenes extends NeRF to dynamic scenes by learning a temporally coherent neural radiance field from a set of multiview images with known camera parameters. D-NeRF models the time-varying appearance and geometry by conditioning the neural radiance field on a latent code that varies with time. The method introduces an additional loss term to enforce temporal smoothness in the learned radiance field, enabling high-quality view synthesis of dynamic scenes with temporal consistency. This approach enables the reconstruction of dynamic scenes from multi-view videos and synthesizes novel views while preserving appearance and motion continuity [Pum+20].

NeRFacto: Nerfacto is a method developed specifically for the Nerfstudio framework. It is a 3D scene reconstruction and rendering method that builds on the structure of MipNeRF360 [Bar+21] and incorporates ideas from other advanced techniques like NeRF– [Wan+21], Instant-NGP [Mül+22], NeRF-W [Mar+20], and Ref-NeRF[Ver+21]. This integration enhances performance, efficiency, and visual quality, making Nerfacto a robust and optimized method for realistic scene rendering [Tan+23].

LERF: Language Embedded Radiance Fields optimizes a dense, multi-scale language 3D field by volume rendering CLIP embeddings along training rays, supervising these embeddings with multi-scale CLIP features across multi-view training images. After optimization, LERF can extract 3D relevancy maps for language queries interactively in real-time. LERF enables pixel-aligned queries of the distilled 3D CLIP embeddings without relying on region proposals, masks, or fine-tuning, supporting long-tail open-vocabulary queries hierarchically across the volume [Ker+23].

2.3 NeRF Frameworks Overview

Instant NGP is a NeRF platform created by NVIDIA Labs from the paper Instant Neural Graphics Primitives with a Multiresolution Hash Encoding [MÜL+22]. Neural graphics primitives, parameterized by fully connected neural networks, are often costly to train and evaluate. This cost is reduced through a versatile new input encoding that allows for the use of a smaller network without compromising quality, significantly cutting down on floating point and memory access operations. A small neural network is enhanced with a multiresolution hash table of trainable feature vectors, optimized via stochastic gradient descent. The multiresolution structure enables the network to resolve hash collisions, resulting in a simple architecture that is easily parallelized on modern GPUs. This parallelism is fully utilized by implementing the system with fully-fused CUDA kernels, focusing on minimizing wasted bandwidth and compute operations. The result is a combined speedup of several orders of magnitude, allowing for the training of high-quality neural graphics primitives in seconds and rendering in tens of milliseconds at a resolution of 1920×1080 [MÜL+22].

ArcNeRF is a modular framework developed by Tencent's Applied Research Center Lab for novel view rendering and object extraction based on NeRF. It integrates various state-of-the-art NeRF-based methods and allows easy modification and development of custom algorithms. The framework is highly modular, allowing users to modify any component in the pipeline and develop their own algorithm easily. The framework includes features such as modular pipeline design, mesh extraction, surface rendering, and compatibility with datasets and benchmarks. It supports NeRF and Neural Implicit Surface (NeuS) models with various extensions such as volume pruning and hash grid embedding. The project aims to facilitate research and development in 3D scene rendering [TEN23]. Nevertheless, its complex installation process and the considerable learning curve associated with training on user-generated datasets it apart from nerfstudio.

Nerfstudio is a modular framework for NeRF development and was introduced by Tancik [TAN+23] to streamline NeRF research and application. This Python-based framework, designed at the University of California, Berkeley, integrates various NeRF techniques into reusable components, enabling real-time visualization and simplifying the workflow for user-captured data. Therefore Nerfstudio is the largest and most well-known framework in this realm.The framework's modularity allows for the abstraction of method-specific implementations, making it adaptable for different models and data input formats. This flexibility consolidates these research innovations and community-driven development making nerfstudio particularly beneficial for incorporating real-world scenes captured by users. Even offering its own customized training method, 'Nerfacto,' described earlier.

Figure 2: Nerfstudios Live Web Viewer: Displays various scene configuration options and showcases the camera inputs used within the scene.

Image source: [Wid24]

The framework's real-time web viewer, crucial for qualitative evaluations, supports interactive visualizations during training and testing. This viewer is instrumental in assessing model performance, especially for novel views and unstructured environments. The nerfstudio Dataset, comprising real-world scenes, further aids in benchmark NeRF methods. The framework's open-source nature, with contributions from both academia and industry, underscores its potential for community-driven development in neural rendering. For this reason, nerfstudio will be used as an example framework for this research [Tan+23].

3 Detailed Overview of the Nerfstudio Framework

Nerfstudio provides a flexible and comprehensive framework for the development of NeRFs. The objective is to consolidate disparate NeRF techniques into reusable, modular components, thereby facilitating real-time visualization of neural radiance field scenes. The user-friendly controls provided by Nerfstudio facilitate an end-to-end workflow for the generation of neural radiance fields from both popular NeRF datasets and user-generated data. The design of the library facilitates the implementation of NeRF by dividing each element into modular units, thereby enhancing interpretability and user experience [Tan+23].

3.1 Nerfstudio Installation

The objective of this section is to illustrate the generation of user-defined 3D models using Nerfstudio, trained with different Nerfstudio methods, 'Nerfacto' and 'Lerf', which are based on distinct NeRF models. Both methods facilitate the generation of versatile 3D models that can be employed in a multitude of applications, including the creation of animated path rendering. The result can simulate the experience of flying around the scanned object with a small drone. Furthermore, the models can be exported as meshes for additional processing in software such as Blender. Given the frequent modifications to the installation procedures and versions, these instructions are designed to offer comprehensive and detailed assistance.

The following system specifications were used:

Component	**Specifications**
Laptop Model	Lenovo Legion Pro 7 16IRX8H (82WQ)
Processor	Intel Core i9-13900HX, 3.9 GHz
Operating System	Windows 10 Pro
Graphics	NVIDIA GeForce RTX 4080 Laptop GPU
Memory	32 GB RAM 5600 MHz
Storage	1 TB SSD NVMe

Table 1: System Specifications

In order to ensure the autonomy of our work and its potential for independent replication, we have chosen to utilize the Docker platform.

Docker provides the capability to package and execute an application within a loosely isolated environment, termed a container. The isolation and security afforded by containers enables the concurrent execution of multiple containers on a single host. Containers are lightweight and encapsulate the entire runtime environment necessary for an application to run, eliminating the need to rely on the host's pre-installed software. Containers can be shared while working on a project, ensuring that all collaborators utilize the same container with the same configuration [Doc24].

3.1.1 Installing WSL

To begin, we will install WSL (Windows Subsystem for Linux), which provides a Linux environment directly within Windows. Detailed installation instructions can be found in the official Microsoft documentation [Mic24].

1. **Install WSL**: Open PowerShell and run the following command to initiate the installation:

```
wsl --install
```

2. **Verify Installation**: After the installation completes, verify that WSL is correctly installed by running:

```
wsl -l -v
```

   This command lists the installed Linux distributions along with their version information. In our setup, Ubuntu and two Docker environments are installed and operational.

3.1.2 Installing Nvidia Drivers

The next step is to install the Nvidia drivers in order to guarantee the optimal utilization of the graphics processing unit (GPU).

1. **Download Drivers**: Visit the Nvidia website and select the appropriate product series, product family, and operating system. Download the latest drivers for your GPU.
2. **Install Drivers**: Run the installer to update your drivers. Since our system already has the latest drivers, we will skip this step.

3.1.3 Installation of Docker Desktop

The installation of Docker Desktop is a prerequisite for containerizing the Nerfstudio environment.

1. **Download Docker Desktop**: Navigate to the official Docker website and select "Docker Desktop for Windows" to download the installer.
2. **Install Docker Desktop**: Follow the on-screen instructions to complete the installation. If Docker Desktop is already installed on your system, this step can be skipped.
3. **Verify the Installation**: Launch Docker Desktop and confirm that the Containers, Images, and Volumes sections are empty, which indicates a successful and clean installation.

3.1.4 Troubleshooting and Solution Guide for Docker Pull

Instead of installing and compiling prerequisites, setting up the environment and installing dependencies, a ready-to-use Docker image is provided by Nerfstudio.

1. **Pull Docker Image**: Execute the following command to download a specific version of Nerfstudio:

```
docker pull dromni/nerfstudio:<version>
```

This process may also take some time.

2. **Run Docker Container**: Use the following command to run the Docker container with GPU support and mounted volumes:

```
docker run --gpus all -v C:/Users/DominikW/NERF/nerfstudio_workspace:/workspace/ -v
    C:/Users/DominikW/NERF/nerfstudio_workspace_cache:/home/user/.cache/ -p
    7007:7007 --rm -it dromni/nerfstudio:main
```

During our work, we encountered significant challenges in pulling and executing a Docker image. Upon conducting a thorough analysis, we determined that the image was likely constructed with certain shared libraries that incorporate an outdated version of the NVIDIA library.

The primary issue appears to be a version mismatch between the shared libraries within the container and the CUDA Toolkit installed on our system. Typically, the necessary NVIDIA software is expected to be injected from the host system into the container at runtime, facilitated by the '–gpus all' flag. This process should ensure that the correct version of the library is utilized, consistent with the host system's configuration. However, in this instance, the version conflict has resulted in the image failing to function as intended. For further information, please refer to the following sources: NVIDIA-CONTAINER-TOOLKIT/ISSUES/289 and NVIDIA-CONTAINER-TOOLKIT/ISSUES/289.

To mitigate this issue, we explored the use of an older NVIDIA driver. By reviewing the release date of the Docker image, we determined that it was published on May 13, 2024. At that time, NVIDIA driver version 552.44 was the most current, although it is possible that the image was built using the slightly older version 552.22. As a corrective measure, we opted to employ the image with version 1.1.0 rather than the current "main" tag. This approach allows us to fix the version, thereby preventing any unexpected changes from future updates.

Despite numerous modifications, we were unable to utilize the image in the manner originally intended. Consequently, we decided to build the image ourselves, therefore please refer to chapter 3.2.1 *"Docker Build.*

3.2 Download and Installation of Nerfstudio

To begin, we will install the Nerfstudio project from GitHub.

1. **Download Nerfstudio**:
   (a) Open your web browser and navigate to the Nerfstudio project directory on GitHub.
   (b) Once on the GitHub page, click on the "Code" button.
   (c) From the dropdown menu, select "Download ZIP" to download the entire project as a ZIP file.
2. **Extract Files**:
   (a) Locate the downloaded ZIP file on your computer.
   (b) Right-click on the ZIP file and select "Extract All..." or use a similar extraction tool.
   (c) Choose a suitable location on your computer for the extracted files. In this guide, we will extract the files to the following directory:

```
C:\Users\DominikW\NERF
```

   (d) After extraction, the entire Nerfstudio project should be available in the chosen directory.

3.2.1 Docker Build

Furthermore, difficulties have been encountered when constructing the container independently, which can be attributed to the necessity of implementing alterations within the Docker file. The requisite adjustments are provided in the attachment for your convenience.

1. **Navigate to the Project Directory**: First, open your terminal and navigate to the directory where the Nerfstudio project is located. The command is:

```
cd C:\Users\DominikW\NERF\nerfstudio-main
```

2. **Build the Docker Image**: Within the project directory, you will find a 'Dockerfile', which serves as the foundation for compiling the Docker image. To build the image, use the following command:

```
docker build --no-cache \
    --build-arg CUDA_VERSION=11.8.0 \
    --build-arg CUDA_ARCHITECTURES=89 \
    --build-arg OS_VERSION=22.04 \
    --tag nerfstudio-01 \
    --file Dockerfile .
```

   Note: It is crucial to set the correct CUDA architecture if your hardware requires it. For example, if you are using a RTX 2000 series, you should set the architecture to 75 by including the appropriate flag in the build command. For further information, please refer to the following source: Matching CUDA arch and CUDA gencode for various NVIDIA architectures.

3. **Wait for the Build to Complete**: The entire build process may take some time, depending on your hardware specifications. Ensure that the process completes successfully before proceeding.

3.2.2 Starting the Docker Container

After the Docker image is built, we can start it as a container.

1. **Run Docker Container**: Use the following command to start the container, passing through all graphics cards and mapping directories within the image to the local drive. Also, specify port 7007 for the viewer:

```
docker run --gpus all -v "C:/Users/DominikW/NERF/workspace:/workspace" -v "C:/Users/
    DominikW/NERF/workspace_cache/.cache:/home/user/.cache" -p 7007:7007 --rm -it
    nerfstudio-01
```

2. **Access Workspace**: After running the command, you will be placed in the ‘workspace‘ directory within the container, from where you can execute various Nerfstudio commands.

3.3 Copying Media

In this step, we will prepare the input data by copying it into the designated workspace directory.

1. **Create the Workspace Directory**:

   (a) On your local computer, open the file explorer or use a terminal to navigate to the location where you want to create the workspace.

   (b) Create a new directory named `workspace` where the input data will be stored. If you are using the command line, you can create the directory with the following command:

```
mkdir workspace
```

2. **Copy Input Data**:

   (a) Gather the input data, which consists of approximately 50 photos taken with your phone. A comprehensive account of the methods employed in the acquisition of data is presented in the chapter 5 on Smartphone-Based Dataset Acquisition.

   (b) Copy these images into the `workspace` directory you just created. If you are using the command line, navigate to the directory containing your images and execute:

```
ns-process-data images --data /workspace/input/robot_blue --output-dir /
    workspace/processed/robot_blue
```

   Note: A comprehensive explication of the procedure can be found in chapter 5 Estimate camera poses from images with COLMAP.

3.3.1 Training with Nerfacto

In this section, we will train the data using the Nerfacto method.

1. **Start Training with Nerfacto**:
   (a) Ensure that you are inside the Docker container where Nerfstudio is set up.
   (b) Execute the following command to initiate the training process:

```
ns-train nerf-facto --data /workspace/processed/robot_blue
```

   (c) This command starts the training process using the Nerfacto method. The data located in the `/workspace/processed/robot_blue` directory will be used for generating the NeRFs.

2. **Performance Considerations**:
   (a) Note that while the training times for Nerfacto are comparable to other methods, it requires significantly more computational power during live previews. Therefore, we recommend that you use the Viewer only in selected cases during training. This can speed up training and improve the user experience in the Viewer.
   (b) Ensure your system is capable of handling the increased load, especially if you plan to interact with the live preview during training.

3.3.2 Connecting to the Viewer

The Nerfstudio viewer is launched automatically with each `ns-train` training run. However, it can also be run separately using the `ns-viewer` command.

1. **Accessing on a Local Machine**:
   (a) Once the viewer is running, you can open it in your web browser by entering the server's address and port.
   (b) Typically, this will look like `http://localhost:7007`. If you are running the viewer locally, simply click the link provided when you run the script to open it in your browser.

3.3.3 Creating and Exporting the Flythrough

1. **Access the Render Tab**:
   (a) Open the viewer where your 3D model is displayed.
   (b) Navigate to the *Render* tab within the viewer interface.
2. **Add Keyframes**:
   (a) Use the *Add Keyframe* button to define specific points in the 3D space. These keyframes will determine the path that the virtual camera will follow.
   (b) Carefully position the keyframes to capture the desired angles and movements within the 3D environment.
3. **Render the Flythrough**:
   (a) Once all keyframes are set, initiate the rendering process. The rendering may take some time depending on your hardware specifications.
   (b) The rendered video will be saved in the `workspace` directory. Ensure there is sufficient space and that the rendering process is not interrupted.
4. **Save the Video**:
   (a) After rendering, the final video will be automatically saved in the following directory: `workspace/renders`
   (b) You can now access the rendered video from this location for further use or distribution.

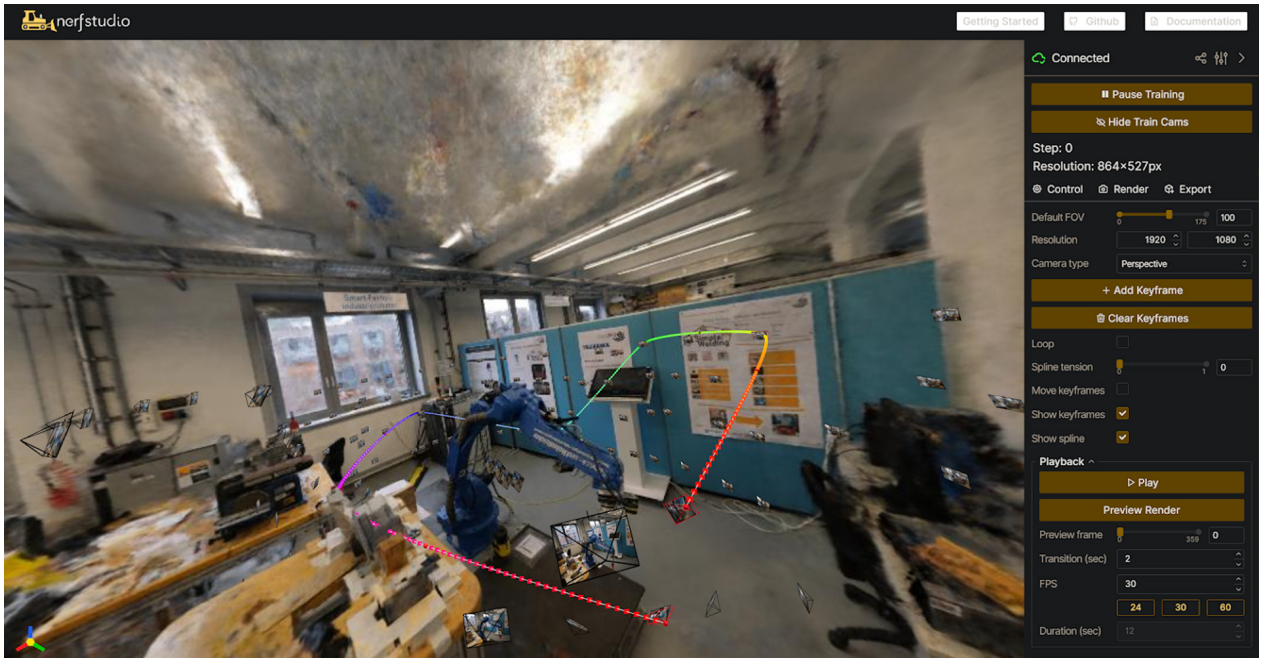


Figure 3: Overview of possible settings for creating a camera animation path (highlighted) in the Nerfstudio Viewer: Selecting cameras within the scene has proven to be extremely helpful for easily creating a camera path, allowing for quick and precise determination of the viewing direction.
Image source: [Wid24]

4 Smartphone-Based Dataset Acquisition

4.1 Introduction to Smartphone-Based Dataset Acquisition

The utilization of NeRF-based applications initially requires the availability of suitable hardware and software for data acquisition, pre-processing, training, and visualization. Given the interconnected technical processes, the first step is to generate technically flawless datasets as input data - No data is better than bad data. To achieve this, we present a guideline for the correct acquisition of datasets, allowing smartphones to be used directly on-site without pre-calibration. In this section, the potential camera settings of smartphones as data acquisition devices are compared to compact cameras based on their results when creating NeRF 3D representations. For this purpose, two typical industrial environments were captured under conditions as close to reality as possible.

Understanding the tools available for use as acquisition devices is fundamental to this work, requiring an examination of individual camera components, such as those found in SPC and mirrorless system cameras, to determine their impact on outcome quality. To comprehend how these components influence the resulting images, it is essential to consider and understand each element of the cameras used. Given that SPCs are the most widely used, they will serve as the primary reference point for this explanation. Specifically, we will refer to the camera model of the Huawei P30 Pro smartphone, which was utilized for some of the shots in this study. The objective is to provide the knowledge required to take sharp, depth-of-field images with a smartphone camera that meet the requirements for NeRF pre-processing and NeRF training.

In this study, a smartphone (Huawei P30 Pro) and a mirrorless compact camera (Fujifilm X-T5) were utilized to capture the input data. The specifications of both cameras are provided in Table 2.

Specification	**Mirrorless Compact Camera (Fujifilm X-T5)**	**Smartphone (Huawei P30 Pro)**
Camera model	Fujifilm X-T5	VOG-L29
Sensor type	X-Trans CMOS	BSI-CMOS
Sensor size	APS-C (23.5 mm × 15.6 mm)	1/1.7” (7.53 × 5.65 mm)
Camera resolution	40.2 MP	40 MP + 20 MP + 8 MP
Pixel size	3.76 µm	˜1.0 µm
Focal length	12 mm (Equivalent: 18 mm)	18 mm – 135 mm
Crop factor	1.5	5.64
ISO range	125 – 51,200	50 – 409,600
Aperture	f/2.0	f/1.6 – f/3.4
Image format	HEIF, JPG, RAW (RAF)	JPEG, DNG

Table 2: Specifications comparison between Fujifilm X-T5 and Huawei P30 Pro

4.2 Principles of Light and Imaging

Eyes are functionless without light; and so are cameras. Light is the central element responsible for illumination and reflection. Cameras operate on that principle. A camera captures light reflected into it within its field of view. In most cases, however, light is scattered in all directions simultaneously. To ensure the capture of precise and useful light information, rather than a blurred image, a camera lens is required. The camera lens functions by converging divergent light rays, which are scattered in various directions, and redirecting (refracting) them to converge at a single, focused point on the image sensor. This focused point allows the camera sensor to accurately record the light information [Rus21].

Early cameras used film, a light-sensitive material that changes chemically based on the amount of light that passes through the lens and strikes it. If the film is exposed to light for extended periods, it becomes fully exposed and appears white due to the complete chemical reaction. To control the exposure time and avoid overexposure, cameras use a shutter that opens and closes rapidly to allow the correct amount of light to hit the film. In modern digital cameras, the film is replaced by an image sensor (CMOS (Complementary Metal-Oxide-Semiconductor), CCD (Charge-Coupled Device))[BSI], which electronically reacts to light. Unlike film, the sensor can capture images repeatedly, converting light into digital signals for processing and storage.

In digital cameras with electronic exposure, the sensor itself regulates the exposure duration electronically. Instead of relying on a physical mechanical shutter to control when light hits the sensor, the electronic sensor initiates and terminates light capture electronically. The sensor determines the precise moment to begin and end light collection, effectively acting as its own shutter mechanism. This capability allows for manual fine control over exposure times. The exposure time or shutter speed is specified in seconds, typically presented as fractions, such as 1/30 or 1/400 of a second.

Proper exposure control is crucial for capturing details in a scene, which significantly impacts the quality of datasets used for NeRF applications, particularly when pre-processed with Structure From Motion (SFM) tools like COLMAP. Proper manual exposure control ensures that the captured images have a balanced ratio between light and shadow, preserving details across the entire dynamic range of the scene [Man+23]. Overexposure can lead to the loss of details in bright areas, while underexposure can obscure details in dark areas.

Capturing these details is essential for the extraction of features, which are specific, distinct points or patterns within an image that SFM algorithms use for various tasks. In computer vision, features derived from well-exposed images are crucial for geometric reconstruction, bundle adjustment, visual odometry, stereo correspondences, visual SLAM (Simultaneous Localization and Mapping), bag of words approaches, and loop closing. Ensuring that images are correctly exposed allows these features to be accurately detected and described, enhancing the performance and reliability of these algorithms. Proper exposure control directly influences the ability to capture and utilize these essential details, ultimately impacting the effectiveness of computer vision applications [Ope].

Sufficient exposure can be verified at the time of capture with the aid of photographic histograms. With the help of a histogram covered in detail in chapter 4.3.2, we can assess how the tonal values, such as the brightness or contrast, are distributed across the image, even without seeing the image if required. This is particularly helpful in case of direct sunlight on the smartphone display, as beyond a certain brightness, it is not immediately noticeable whether all areas of the scene are sufficiently represented. Also, in some cases, the perceptual representation of OLED (organic light-emitting diode) displays is not sufficient to read the required information directly. An image histogram is a visual representation that demonstrates the frequency of occurrence of different intensity values within an image. It can be used to indicate the proportion of instances where specific colours, such as black, white, and shades of grey, are present in an image. Additionally, the camera exposure control can be adjusted to affect various aspects of the image, including contrast and brightness [24c].

When creating datasets, shutter speed or exposure time is crucial for capturing sharp images and avoiding motion blur. The following guidelines can help achieve this. It is recommended to use a shutter speed that is at least the reciprocal of the focal length to avoid camera shake. This means the shutter speed should be fast enough to prevent blurring by being at least 1 divided by the focal length of the lens used (focal length of 50 mm, shutter speed 1/50) second to ensure sharp images. Additionally, using faster shutter speeds (e.g., 1/125 second or faster) can help avoid motion blur while movement during shooting, while slower shutter speeds (e.g., 1/60s or slower) are suitable for low light conditions a tripod or any other fixed device can be used. Therefore, the selected camera settings determine the outcome of the photograph. When time is limited, the user must act quickly and add lighting in low-light conditions to ensure proper exposure.

Figure 4: Camera with Focus Peaking enabled: This feature assists in real-time by highlighting areas in focus, allowing you to verify that all settings are correctly chosen for capturing the scene sharply and accurately during the shot.

Image source: [Wid24]

SPCs often feature multiple lenses, which function similarly to interchangeable lenses on traditional cameras. Each lens on a smartphone offers a different viewpoint and focal length, allowing for versatility in capturing various scenes. This multi-lens setup allows multiple lenses without swapping them, providing the convenience of different perspectives within a single device. Focal length is a key specification in photography and is typically indicated in millimeters (mm). It represents the distance between the lens and the image sensor (e.g., 50 mm indicates a fixed focal length lens (prime lens), while a range e.g., 18-55 mm indicates a zoom lens that can vary between different focal lengths).

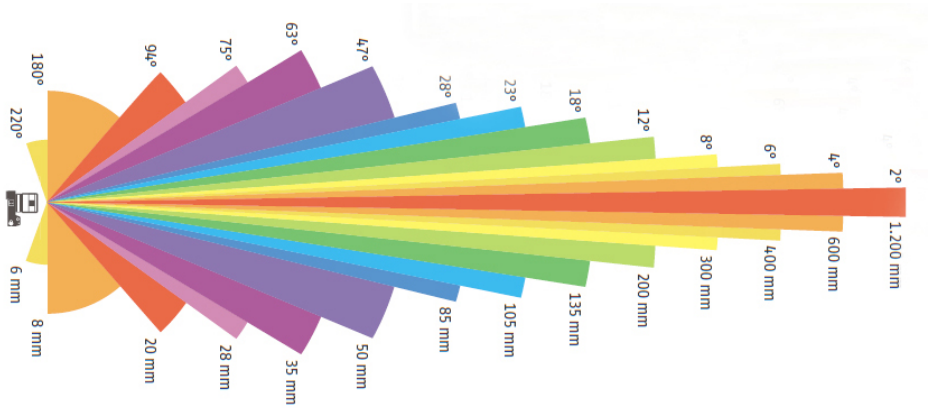


Figure 5: Focal Length: Defines the field of view in your image. A shorter focal length (lower millimeters) gives a wider angle, capturing more of the scene. A longer focal length (higher millimeters) narrows the angle, focusing on a smaller part of the scene. Short focal lengths are ideal for wide-angle shots, while longer ones are suited for telephoto images.

Image source: [LL23]

The Huawei P30 Pro features several camera systems with multiple lenses, each developed for specific photographic purposes:

Camera Type	**Resolution**	**Sensor**	**Focal Length**	**Aperture**	**OIS**
Main Camera	10 MP (40 MP)	1/1.70”	27mm	F1.6	Yes
Wide Angle	20.0 MP	1/2.70”	16mm	F2.2	No
Telephoto	8.0 MP	1/4”	125mm	F3.4	Yes

Table 3: Technical Specifications of the Huawei P30 Pro Cameras

Multiple independent camera systems in a confined space necessitate having three separate camera sensors. This means that each individual sensor must be smaller, resulting in poorer light sensitivity. Therefore, the main camera of the P30 Pro uses a process known as pixel binning, where four pixels are combined into one (ultra) pixel by default. This reduces the resolution but enhances image quality, resulting in a resolution of 10 megapixels. Understanding this is essential for calculating critical values such as the Ground Sampling Distance (GSD), further elaborated in the chapter on datasets.

The ultra-wide-angle camera of the Huawei P30 Pro features a 20 MP resolution, a 16 mm focal length, and an aperture of f/2.2. This configuration provides an expansive field of view, ideal for capturing wide scenes such as architectural interiors. The f/2.2 aperture balances light intake and depth of field, which is the area of an image that appears sharp. Offering a broader depth of field than the main camera, making it particularly useful for detailed wide-angle shots. To effectively capture the features of a scene, multiple contiguous images from several camera positions are necessary. By analyzing these images, software calculates the positions of features by determining the angles and distances between the camera positions, forming a triangulation. For instance, when mapping a building to create an orthomosaic (The output from a process where a number of overlapping photos are stitched together with distortions removed to create a complete and continuous image representation [24D].), it is crucial to conduct the imaging process in a structured manner and ensure that successive images overlap. This overlap is much easier to achieve with a wide-angle lens, making the ultra-wide-angle camera particularly advantageous. This aspect will be further elaborated in the chapter on datasets under the section 5.1 *'Overlap'*.

While the dynamic range and colors of the ultra-wide-angle camera in JPEG files are impressive, its overall sharpness, especially towards the edges, does not match that of the main camera. This phenomenon can be attributed to the fact that the optomechanical design and manufacturing technology of smartphone lenses differ significantly from those of traditional camera lenses. Factors such as the use of highly aspheric plastic lenses, miniaturization, and the fully automated production of millions of units make them incomparable to traditional optical manufacturing [BS21].

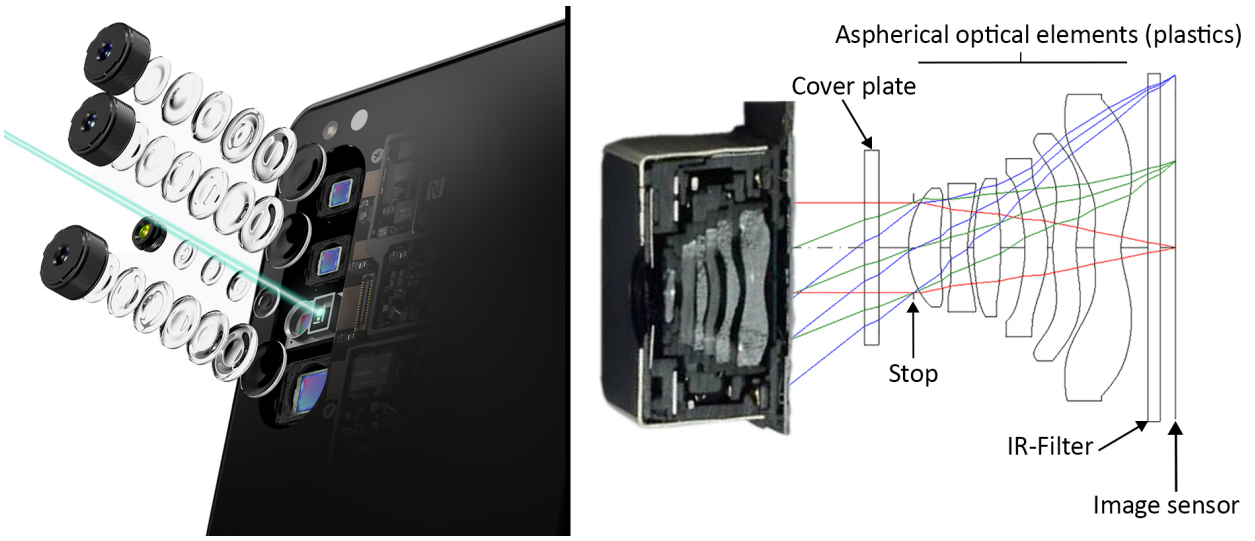


Figure 6: Structure of a wide-angle lens: The light coming from a faraway object enters the lens at a cover glass (approx. 0.2 mm thick), then passes through the plastic lens elements and an infrared (IR) filter (thickness approx. 0.2–0.3 mm) before finally arriving at the image sensor. The fixed lens stop is usually placed at the lens entrance[BS21].

Image source: [BS21]

This discrepancy in sharpness can be attributed to several factors:

1. **Lens Aberrations:**
    - **Chromatic Aberration:** This occurs when different wavelengths of light are refracted by varying amounts, causing color fringing and a loss of sharpness at the edges of the image.
    - **Spherical Aberration:** Imperfections in the lens shape can cause light rays to focus at different points, leading to a blurry image.
2. **Diffraction:** When the aperture is very small, diffraction can cause light to spread out and interfere, reducing the overall sharpness of the image. This effect is more pronounced in smaller sensors like those in smartphones.

A potential solution to the sharpness issues commonly encountered with wide-angle smartphone lenses is the adoption of higher-quality lenses designed to reduce aberrations, distortion, and other optical imperfections.

Optical image stabilization (OIS) is a subcategory of image stabilization (IS), which encompasses mechanisms designed to enhance image and video capture by reducing motion blur and shaking effects. OIS, commonly used in SPCs, utilizes a floating lens, gyroscopes, and small motors to control the path of light to the image sensor. These small motors move the image sensor in the opposite direction of any movements occurring during capture, allowing the sensor to be more precisely exposed. IS includes a broader range of techniques, both hardware- and software-based. Hardware solutions, such as in-camera or in-lens stabilization, are typically used for still images, while software algorithms are employed for video stabilization either in real-time or during post-processing. While IS is beneficial for general photography and videography, it can have negative effects on pre-processing with SfM software. A more detailed analysis of this subject can be found in [NMV22]. Preliminary findings indicate that IS should be disabled when accuracy is crucial in 3D scene reconstruction.

The aperture controls the amount of light entering the camera, which directly influences other exposure parameters such as ISO and shutter speed. The aperture size is expressed in f-numbers or f-stop numbers. A smaller f-number indicates a larger aperture, while a larger f-number corresponds to a smaller aperture. A larger aperture (smaller f-number) allows more light into the camera, enabling the use of faster shutter speeds or lower ISO values. Conversely, a smaller aperture (larger f-number) restricts the amount of light entering the camera, necessitating adjustments in ISO and exposure time to maintain proper exposure [The]. ISO sensitivity, f-stop numbers, and exposure time are all scaled logarithmically to base 2. ISO values double in a sequence such as ISO 100, ISO 200, ISO 400, etc. F-stop values increase by factors of the square root of 2, such as f/2, f/2.8, f/4, etc., with each stop representing a halving or doubling of the light intensity. Exposure times follow a sequence like 1/100, 1/50, 1/25 seconds, etc. According to the 'sunny 16 rule,' increasing the aperture by 3 f-stops to f/5.6 allows for an eightfold reduction in exposure time or a corresponding adjustment in ISO.

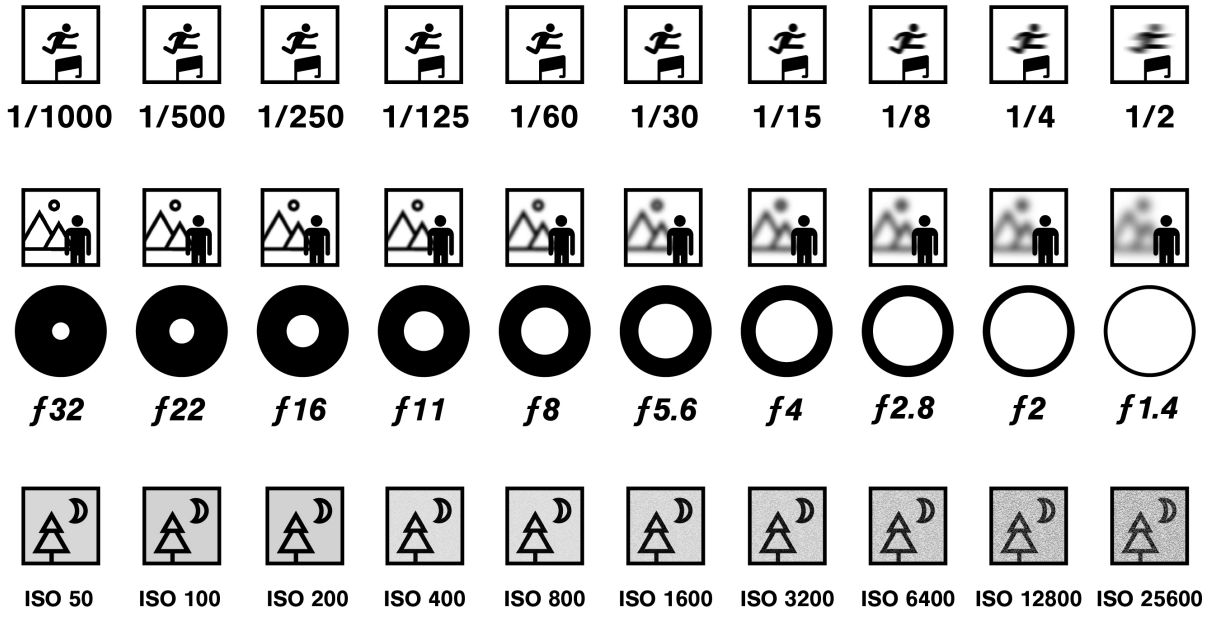


Figure 7: Illustrating the effects of aperture, shutter speed, and ISO: Aperture controls depth of field, shutter speed affects motion blur, and ISO adjusts the image's sensitivity to light, influencing noise levels.
Image source: [Cam]

In the context of SPCs, however, the aperture is a fixed component in most models, and its effects are mainly understood in relation to other parameters:

- **Sensor Size:** The small sensors used in smartphones provide a greater depth of field, which allows more of the scene to remain in focus even at wide apertures like f/1.8 or f/2.8. It applies: a smaller sensor size leads to a greater depth of field.
- **Focal Length:** The short focal lengths typical in SPCs further enhance depth of field, ensuring sharp focus throughout the image. It applies: a shorter focal length results in a greater depth of field. (Nevertheless, the changes in depth of field are not linear; for example, halving the focal length will more than double the depth of field.)
- **Aperture:** A smaller aperture equals larger f-number leads to a greater depth of field.
- **Distance:** A greater distance to the subject increases the depth of field.
- **Computational Photography:** Techniques such as image stacking and software algorithms are employed to extend depth of field either the range of brightness or luminance values that can be represented in an image or scene (High-Dynamic Range(HDR)).
- **Software Depth Mapping:** Depth sensors like Time of Flight (ToF) enable post-processing to maintain focus on desired areas while blurring others, mimicking the effects of larger sensors and variable apertures.

This implies that smartphones are theoretically incapable of producing a shallow depth of field (e.g., bokeh effect) on their own, and therefore it is our responsibility to disable effects achieved with aids such as ToF sensors.

Furthermore aperture size directly impacts the depth of field illustrated in figures 8 and 9. This relationship is governed by the principles of optical physics, where the aperture size affects the circle of confusion, which in turn determines the perceived sharpness across different planes of the image. A large f-number (e.g., f/16) will bring all foreground and background objects into focus, whereas a small f-number (e.g., f/1.4) will isolate either the foreground or background, rendering everything else blurry. Referring to the need to effectively capture the features of a scene, it is therefore very important to select the right aperture settings so that the depth of field is maximized. This allows image details to be captured both close up and, in the distance, which is essential for accurate feature extraction.

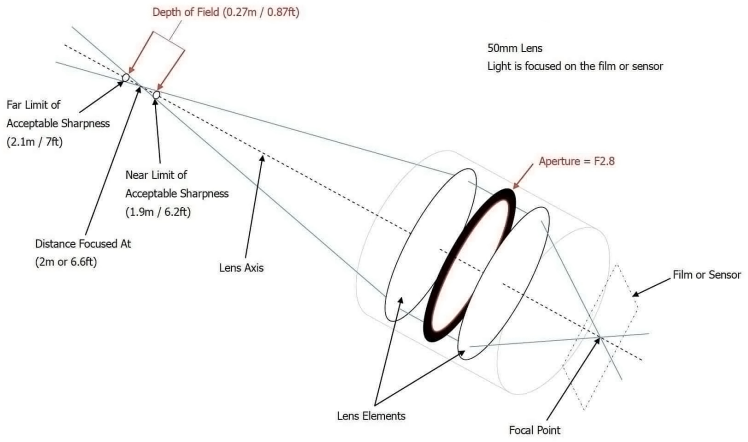


Figure 8: The illustration demonstrates the depth of field with a 50mm lens, with the aperture set at f/2.8 and focused at a distance of 2 meters. Due to the wide aperture setting of f/2.8, the light is focused at an acute angle between the two widest points of the aperture. Consequently, the circle of light as it passes through the aperture reaches a point of unacceptable sharpness relatively quickly. As illustrated in the diagram, the near limit is 1.9 meters, while the far limit is 2.1 meters, resulting in a total depth of field of just 27 cm [Bai21].

Image source: [Bai21]

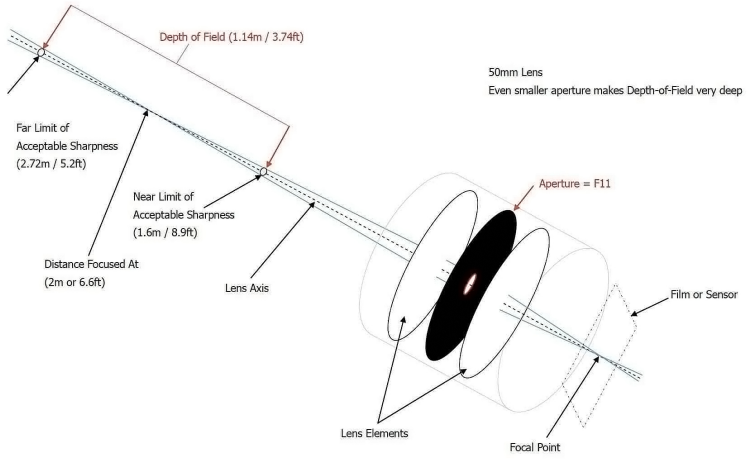


Figure 9: The diagram illustrates an aperture of f/11, which results in a depth of field of 1.14 meters at the same focus distance. This is due to the fact that the light is passing through a smaller aperture, which allows for the circle of acceptable sharpness to be reached at a greater distance from the point at which the lens was focused. It should be noted that all three of these diagrams were based on a 50mm lens focused at 2 meters.[Bai21]

Image source: [Bai21]

4.2.1 Hyperfocal Distance

The hyperfocal distance is influenced by the combined effects of variables such as focal length (F), aperture (N), and the circle of confusion (C), which is determined by the sensor size. The hyperfocal distance is defined as the distance at which a lens should be focused so that all objects from half that distance to infinity appear acceptably sharp. Therefore, it is the focusing distance that maximizes the depth of field for a given lens setting [24A].

To illustrate the concept, we propose a thin lens model, as shown in Figure 23, which we provide in the attachment for aesthetic reasons. The fundamental principle of a lens dictates that collimated light rays, which are parallel to each other and originate from a source at an infinite distance, converge at the lens's focal point [EDM]. As the light source moves closer to the lens, the corresponding focal point shifts away from its original position, necessitating an adjustment of the imaging plane to maintain focus [24A].

A fundamental concept is the "circle of confusion," which describes the largest blur spot on the sensor or film that can still be perceived as a point of light. In essence, it defines the threshold at which a point of light remains "acceptably sharp" even when slightly out of focus.

The hyperfocal distance is the point at which the cone of light rays from infinity produces a blur circle that exactly matches the size of the circle of confusion. When the lens is focused at this hyperfocal distance, the light rays from infinity, though slightly out of focus, fall within the circle of confusion and are therefore still considered in focus. Similarly, the cone of light rays originating from half of the hyperfocal distance also remains within the circle of confusion [24A].

The following formula enables the calculation of the hyperfocal distance for data-set capturing using the Huawei P30 Pro with the focus mode set to infinity.

$$H = \frac{F^2}{N \cdot C}$$

where:

- H is the hyperfocal distance.
- F is the focal length of the lens.
- N is the f-stop number (aperture setting), expressed as $\frac{F}{\text{Diameter of the aperture}}$.
- C is the circle of confusion.

The results for the Fujifilm X-T5 are presented in tabular form.

Aperture	FUJINON XF 35mm f/2 R WR	SAMYANG 12mm f/2.0 NCS CS X
f/2.8	21.91 m	2.58 m
f/4	15.35 m	1.81 m
f/5.6	10.97 m	1.30 m
f/8	7.69 m	0.91 m
f/11	5.60 m	0.67 m
f/16	3.86 m	0.46 m

Table 4: Hyperfocal distance (in meters) for the Fujifilm X-T5 with the lenses FUJINON XF 35mm f/2 R WR and SAMYANG 12mm f/2.0 NCS CS X lenses at different apertures.

Figure 10: Example of a shot where all parameters were chosen to achieve infinite depth of field: The 12mm lens ensures that even objects close to the camera are in sharp focus.
Image source: [Wid24]

Note: In the absence of a variable physical aperture in the Huawei P30 Pro, an approximate f-number was assumed as a reference point. However, it is important to note that the physical effects typically associated with an aperture are largely simulated by software-based image processing. In the practical test, the hyperfocal distance could be approximated to o.9 m for the main lens and 0.40 m for the ultra-wide angle lens.

4.2.2 Grond Sampling Distance

A supplementary condition arises from the hyperfocal distance, which plays a pivotal role in the acquisition of input data. The Grond Sampling Distance (GSD), it serves as a critical parameter in assessing the precision with which surfaces and features are detected and documented [DJI].

The GSD indicates how much area is captured in one pixel. For example, a GSD of 0.5 mm means that one pixel on the image represents 0.5 linear millimeters on the ground. Therefore, one pixel corresponds to 0.25 mm^2 (0.5 x 0.5 mm) in reality. If you want to achieve a certain GSD, you must know both the camera model (sensor size, image width, focal length) and the exact flight height (if captured by a drone) or distance to the captured object when taking the picture and optimize it accordingly [24B].

The Ground Sampling Distance (GSD) can be calculated using the following formula:

$$\text{GSD} = \frac{\text{Sensor size in mm} \times \text{Distance in m}}{\text{Image size in pixels} \times \text{Focal length in mm}}$$

Information for the Huawei P30 Pro:

- **Main Camera:**
  - Sensor size: 1/1.7" (approximately 7.53 mm x 5.64 mm)
  - Image size: 40 MP (7304 x 5472 pixels)
  - Focal length: 27 mm (Equivalent)
- **Wide-angle Camera:**
  - Sensor size: 1/1.54" (approximately 8.1 mm x 6.1 mm)
  - Image size: 20 MP (5120 x 3840 pixels)
  - Focal length: 16 mm (Equivalent)

Calculation of GSD: For the main camera and the wide-angle camera at distances of 1.5 m and 2.3 m:

- **Main Camera at 2.3 m:**

$$\text{GSD}_{\text{Main, 2.3 m}} = \frac{7.53 \times 2.3}{7304 \times 27} \text{ m/pixel} = \text{ 0,0845 mm/Pixel}$$

- **Wide-angle Camera at 1.5 m:**

$$\text{GSD}_{\text{Wide, 1.5 m}} = \frac{8.1 \times 1.5}{5120 \times 16} \text{ m/pixel} = \text{ 0,148 mm/Pixel}$$

The fundamental premise behind the Ground Sampling Distance (GSD) is rooted in the sampling theorem, which states that a continuous signal can be accurately reconstructed if sampled correctly. According to this theorem, the sampling rate must be at least twice the highest frequency present in the signal, a concept known as the Nyquist rate. This principle is crucial not only in digital signal processing, telecommunications, and multimedia but also in spatial data acquisition. By adhering to the sampling theorem, continuous signals can be digitized without loss of information, enabling precise reconstruction and manipulation of the signal for various applications [Gee23].

In the context of smartphone-based dataset acquisition, the Nyquist rate plays a pivotal role in ensuring that spatial sampling is sufficiently fine to capture all necessary details. The GSD directly impacts this process by determining the resolution of the captured images. To achieve accurate image feature extraction and enable a complete and error-free 3D reconstruction, it is critical that the GSD is small enough to capture the scene's details and that the sampling rate aligns with the Nyquist rate. This alignment ensures that no information is lost during the digitization process, facilitating high-quality data acquisition.

4.3 Dataset Generation and Requirements

4.3.1 Smartphone Settings Aligned with Dataset Acquisition

The enhanced data collection capabilities of smartphones, such as the Huawei P30 Pro, play a crucial role in modern data acquisition. However, the effective utilisation of this data relies on the implementation of NeRF. The NeRF algorithm interprets and reconstructs data by learning from it and adapting to various conditions. For optimal results, the data generated must be consistent, reproducible, and technically error-free. This is especially important for ensuring the correct functioning of SfM pre-processing, which depends on specific requirements during image capture.

- **Camera Settings Consistency:** The consistency of camera settings is vital for the successful triangulation of images by SfM software, which in turn generates clear and detailed 3D scene information. Key settings such as ISO, aperture, shutter speed, and white balance must be optimised at the outset of the shoot, with the focal point of the scene guiding these choices. Once selected, these settings should be maintained throughout the shoot to ensure uniformity across all images.
- **Manual Mode Usage:** To achieve consistent results, the manual mode of the smartphone image-application should be utilised. This mode allows for the precise adjustment of the camera's ISO, aperture, shutter speed, and white balance. If multiple devices are used for capturing the scene, it is crucial that identical camera settings are maintained across all devices to ensure consistency. Make sure you are using the highest quality (JPG-L, RAW), and all strange or extreme compression/enhancement algorithms are disabled.

Recommended Camera Parameters: Based on the information discussed in the previous chapter, the following camera parameters are recommended for optimal data acquisition. Please refer to Figure 11 for an overview of the user interface and settings.:

- **Metering Mode:** The 'Matrix' mode is designed to capture the entire image, making it ideal for subjects that extend across the frame. The entire width of the image is captured and focused evenly.
- **ISO:** To minimize potential alignment issues caused by grain in the image, a low ISO setting is recommended. An ISO setting of 100 is ideal as a starting point. If a higher ISO is necessary due to lighting conditions, it should not exceed 800.
- **Shutter Speed:** To avoid blurry images, the shutter speed should be set to at least 1/160. In low-light conditions where a slower shutter speed is required, it is advisable to slow down the walking movement during the shoot. The shutter speed should be calibrated to ensure the main subject is properly exposed, with movement adjustments made accordingly. These settings should remain constant until the capture process is complete.
- **Focus:** When shooting video, it is essential to keep the object in focus. An imaginary point within the scene can be conceptualized and its trajectory traced. While some sources suggest that the Huawei P30's SPC autofocus (AF-S) is adequate for stationary subjects, our findings indicate that manual focus (MF) helps avoid focus shifts during recording, resulting in much better image quality.

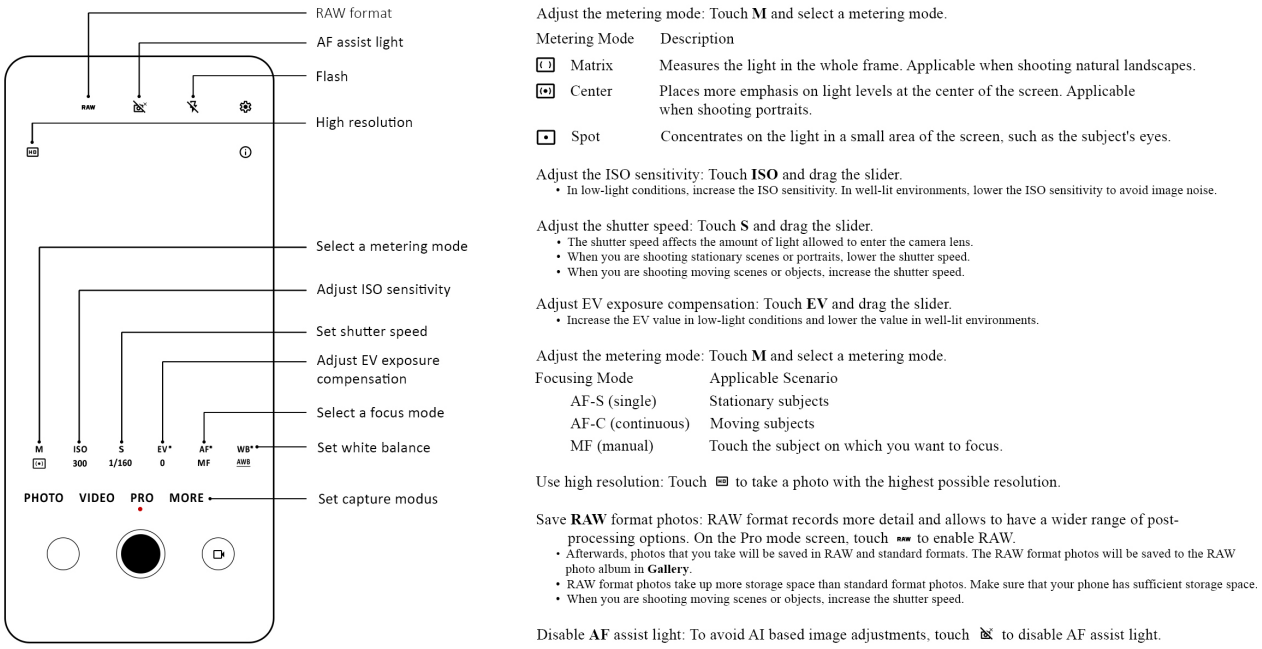


Figure 11: Overview of the UI setting options of the Huawei P30 Pro in manual photo mode [HUA23]. Image source: [Wid24]

4.3.2 Utilizing Image Histograms for Data Validation

Images are not always captured with sufficient quality, which can be due to various factors such as technical issues, time constraints, lighting or recording conditions. However, for the following automated SfM process, the quality of the input images is crucial for achieving accurate and reliable results. To ensure consistent and uniform image quality, image histograms can be employed as a valuable method for assessing image quality. An image histogram is a graphical representation that displays the frequency of different intensity values present in an image, which helps in identifying issues like underexposure or overexposure, thereby aiding in the selection or adjustment of images for SfM processing. In this case, detecting anomalies in histograms is a fast way to assess image quality [24e].

Image histograms provide valuable information that can be used to analyze and enhance images. By applying transformation functions that map current intensity values to new ones, the appearance of an image can be modified. Such transformations are frequently used in photography to make images more visually appealing and are also employed in photogrammetry, computer vision, and other fields to enforce specific properties in an image [Low04].

By adjusting this curve, various aspects of an image, such as contrast and brightness, can be changed. These adjustments can be applied to the entire image or selectively to specific parts, such as shadows, highlights, or mid-tones[Low04].

Histograms can be manipulated through functions known as point operators. These functions take a single intensity value as input and output a corresponding intensity value. For instance, if a function maps an input intensity value of 100 to 120, all pixels with an intensity of 100 will be adjusted to 120. This mapping can be expressed through a curve, often referred to in photography as a tone curve, which indicates how input intensity values are transformed [Low04], [Sta21].

- **Underexposure:** An underexposed image will have a heavily shifted histogram, with a large proportion of pixels concentrated in the dark areas. The middle and right areas of the histogram, representing midtones and bright areas, will be sparse or empty. If images are incorrectly exposed, a number of other anomalies can occur.
- **Overexposure:** An overexposed image will have a histogram that is heavily shifted to the right, with a large proportion of the pixels concentrated in the bright areas. The middle and left areas of the histogram, which represent the midtones and dark areas, will be sparse or empty.

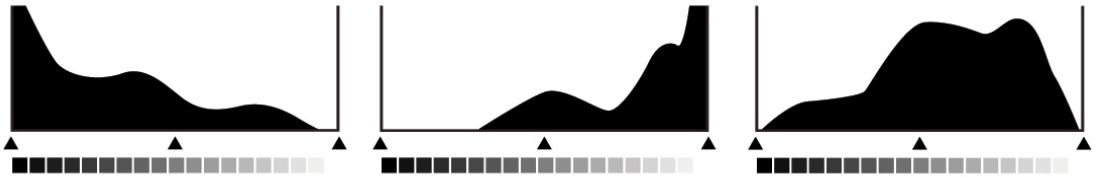


Figure 12: A histogram, in general, is a two-dimensional bar plot where the x-axis represents the elements of interest, such as intensity values or colors, and the y-axis represents the frequency or probability of these elements occurring. In the context of an image histogram, the x-axis denotes the intensity values or colors, while the y-axis shows how many times each color occurs in the image. Comparison of Image Histograms: a) Underexposure, b) Overexposure, and c) Well-exposed.

Image source: [Pea19]

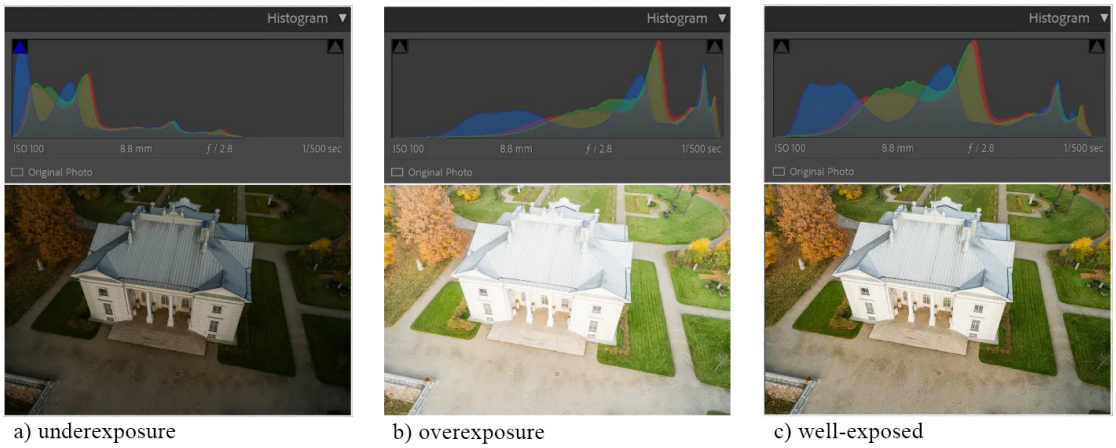


Figure 13: Highlighting the variations in the image caused by different exposures (a) Underexposure, b) Overexposure, and c) Well-exposed): Changes in exposure affect brightness, contrast, and detail visibility of the scene.

Image source: [Pix]

- **Anomalies**
  - **Clipping:** is when a region of your photo is too dark or too light for the sensor to capture any detail. If the histogram is strongly shifted to the left, this may indicate overexposure of the dark areas. This leads to a loss of detail in the shadows, which can be problematic, as fine details are required for SfM.
  - **Narrow and concentrated distribution:** A histogram with a narrow and concentrated distribution, utilizing only a limited portion of the brightness scale, suggests that the image in question exhibits a markedly low contrast. This can impede the ability to discern textures and features.
  - **Multiple peaks:** The presence of multiple peaks in a histogram may be indicative of disparate exposure levels or lighting conditions within the image. This phenomenon can be attributed to the coexistence of both highly illuminated and intensely dark regions, and it should be mitigated to ensure the attainment of uniform images.
  - **Missing Midtones:** The absence of midtones in a histogram may be indicative of an image that is unduly focused on high or low brightness areas, resulting in a paucity of detail in the mid-brightness regions.
  - **Excessive noise:** A histogram exhibiting a markedly uneven or discontinuous gradient may be indicative of image noise. Image noise can compromise the level of detail and the SfM software's capacity to accurately identify and align precise feature-points.

[24e] [Pea19]

4.4 Practical Dataset Acquisition

4.4.1 Objective

The aim of our field trial was to validate the identified requirements for the acquisition of input data and to evaluate their influence on the final result. Our goal is to make NeRF usable for industrial applications in such a way that workers worldwide are able to create high-quality data sets. To achieve this, it is crucial that the data sets collected are of sufficient quality. As part of this test, we carried out an experiment under controlled and documented conditions. The skills of an untrained user (a) were compared with those of a trained user (b). In order to assess the quality of the data produced by the Huawei P30 Pro SPC, two distinct scenarios were defined and the respective environments were recorded using a variety of methods.

4.4.2 Testing

The initial test was conducted in an industrial setting, specifically the Industrial Solutions Laboratory of the Faculty of Industrial Engineering and Management at Furtwangen University. This publicly accessible building contains a multitude of specialized workspaces that are interconnected. The structure and layout are particularly well suited to our purposes, as they allow for the documentation of numerous specific work areas (such as machines, robots, etc.) in addition to the entire area.

Additionally, a comprehensive record was created at the construction site of Furtwangen University's main building, A-Bau. The purpose of this documentation was to capture the construction processes and enable a comparative analysis of their progress over time. Specifically, the construction of an elevator was documented in two distinct phases. Furthermore, the development of several ancillary structures was carefully recorded, providing both an overall perspective and a detailed examination of keypoints of interest. This meticulous documentation aimed to offer a thorough overview of the ongoing construction activities.

The illuminance of the entire scene was measured at multiple intervals throughout the recording sessions. To ensure the comparability of these measurements, they were conducted over several days at similar times of day, with careful documentation of any special circumstances, such as weather-related influences, that could affect the lighting conditions. This approach was intended to maintain consistency in the assessment of lighting across different recording sessions.

4.4.3 Methodology

The first step entailed conducting a practical assessment aimed at evaluating the video recording quality of the Huawei P30 Pro across different environments. The focus of this evaluation was to identify the differences between spontaneous and carefully planned recordings, with particular attention to the stability of the footage and the comprehensiveness of scene coverage.

During the exploratory phase of the investigation, recordings were carried out in an unstructured and unscripted manner to mimic the spontaneous and natural use of the device by an average user (a). The start and end points of the recording were not predetermined, allowing the user to move freely within the scene without adhering to a specific path or returning to a designated point. This approach aimed to evaluate the quality of the recording and the efficiency of image acquisition under unstructured conditions, providing insights into the device's performance in real-world, improvised scenarios.

In the second phase, a structured method based on the concept of "loop clousers" was employed, mimic the capturing process of an trained user (b). It was ensured that the recording was made in a closed loop, which meant that the starting point of the recording was also the end point. The structured approach was designed to minimize errors and ensure comprehensive coverage of the scene. During the filming process, the camera followed a planned trajectory that traversed the scene in a systematic manner. Particular care was taken to ensure smooth movements and a stable image, supported by the use of a gimbal (DJI Ronin SC). We also experimented with different recording techniques.

Technique A: The object was captured using an orbital recording movement while adhering to pre-determined factors such as recording speed, minimum distance, and camera adjustments. Each orbit around the object was performed at a fixed height (190 cm at approximately -25°, 150 cm, 110 cm, 70 cm, and 40 cm at approximately -25°). In the first and last height settings, the camera was tilted downward by about 25° to capture both the upper part of the object and the ground. A folding rule was placed in the scene as a reference for later scale calibration.

Technique B: Similar to Technique A, the object was also recorded using an orbital movement. However, unlike Technique A, the recording heights were not fixed. Instead, the camera was moved around the object in a smooth circular motion. This approach aimed to achieve a balanced capture of the object and ensure a sufficient number of perspectives for analysis.

For both techniques, it should be noted that the external image stabilization (DJI Ronin SC) used encountered issues with the recording methods. The device continuously attempted to interpret and counteract the movement, which, although expected behavior for image stabilization, complicated the planned capture. We attempted to mitigate this by using the device joystick and by adjusting the settings and refining our movements.

4.4.4 Capturing Industrial Solutions Laboratory

The survey of the laboratory building included both the entire laboratory and smaller sub-sections. The clearly structured room layout of the building complex divides it into several subordinate work areas. We used this structure to carry out the so-called "loop closure". Despite the poor weather, there was sufficient natural light due to the open structure of the building, which, in combination with the active lighting, provided approximately sufficient illumination of the room. Nevertheless, several light measurements revealed a relatively low average LUX value. After a comprehensive safety briefing, we examined the individual sub-working areas to identify potential obstacles that could interfere with the image capturing process. The primary aim of the survey was to digitally capture and reconstruct the laboratory and to verify the ability to accurately capture complex, interconnected spaces and work areas. This capture is particularly important to ensure that the digital models accurately reflect both the structure and functionality of the physical spaces. Due to the relatively high light sensitivity of the Huawei P30 Pro, we selected the following settings for the recording: camera mode (manual), focus (MF), shutter speed (medium), ISO (high), metering mode (matrix), stabilization (external).

Property	**XT-5**	**Huawei P30 Pro**
Date	06.07.2024	06.07.2024
Time	1:27 PM - 1:41 PM	1:45 PM - 1:52 PM
Location	Industrial Solutions Lab	Industrial Solutions Lab
Scene	main_corridor	main_corridor
Weather	Cloudy, rainy	Cloudy, rainy
Camera	XT-5	Huawei P30 Pro
Lux (Max./Avg.)	Max. 882 / Avg. 667	Max. 900 / Avg. 680
Settings	ISO 1000, F: 8, SS 1/125	ISO 800, F: 2.2, SS 1/125
Lens	SAMYANG 12mm 1:2.0 NCS CS X	16mm Equivalent (Leica Optics)
Recording Format	4K 29.97 H.264	Full HD 30fps

Table 5: Comparison of XT-5 and Huawei P30 Pro Recording Settings Industrial Solutions Laboratory - Scene: `main_corridor`

Property	**XT-5**	**Huawei P30 Pro**
Date	06.07.2024	06.07.2024
Time	1:55 PM - 2:09 PM	2:14 PM - 2:26 PM
Location	Industrial Solutions Lab	Industrial Solutions Lab
Scene	robot_blue	robot_blue
Weather	Partly cloudy	Cloudy, rainy
Camera	XT-5	Huawei P30 Pro
Lux (Max./Avg.)	Max. 1098 / Avg. 882	Max. 1050 / Avg. 870
Settings	ISO 640, F: 8, SS 1/100	ISO 600, F: 2.2, SS 1/100
Lens	SAMYANG 12mm 1:2.0 NCS CS X	16mm Equivalent (Leica Optics)
Recording Format	4K 29.97 H.264	Full HD 30fps

Table 6: Comparison of XT-5 and Huawei P30 Pro Recording Settings Industrial Solutions Laboratory - Scene: `robot_blue`

4.4.5 Capturing Construction Site Furtwangen University

The recording of the construction site in the main building covered several areas, with the focus on documenting the construction progress of the elevator as well as the recording of a side wing and an exhaust air system. The construction progress of the elevator was recorded in detail, starting with the formwork to the hardened and switched-off state. The condition of a side wing of the building and an exhaust air system were also recorded. The aim of these recordings was to enable before-and-after comparisons and to test models for scene interaction using LERF.

Following a safety briefing, we examined the individual sub-work areas to identify potential obstacles that could impair data collection. The very dark lighting conditions around the elevator posed a particular challenge. To improve these conditions, we used additional lighting, but this failed after a short time. We then tried to adjust the camera settings to compensate for the lighting conditions.

Property	**XT-5**	**Huawei P30 Pro**
Date	22.07.2024	22.07.2024
Time	10:52 AM - 11:07 AM	11:14 AM - 11:29 AM
Location	Construction site A-Bau	Construction site A-Bau
Scene	elevator_unboarded	elevator_unboarded
Weather	Cloudy	Cloudy
Camera	XT-5	Huawei P30 Pro
Lux (Max./Avg.)	Max. 621 / Avg. 423	Max. 518 / Avg. 423
Settings	ISO 1600, F: 6, SS 1/60	ISO 800, F: 2.2, SS 1/60
Lens	SAMYANG 12mm 1:2.0 NCS CS X	16mm Equivalent (Leica Optics)
Recording Format	4K 29.97 H.264	Full HD 30fps

Table 7: Comparison of XT-5 and Huawei P30 Pro Recording Settings Construction Site A-Bau - Scene: `elevator_boarded`

Property	**XT-5**	**Huawei P30 Pro**
Date	22.07.2024	22.07.2024
Time	11:29 AM - 11:42 AM	11:29 AM - 11:42 AM
Location	Construction site A-Bau	Construction site A-Bau
Scene	side_wing	side_wing
Weather	Cloudy	Cloudy
Camera	XT-5	Huawei P30 Pro
Lux (Max./Avg.)	Max. 1423 / Avg. 420	Max. 1219 / Avg. 410
Settings	ISO 1600, F: 6, SS 1/60	ISO 1000, F: 2.2, SS 1/50
Lens	SAMYANG 12mm 1:2.0 NCS CS X	16mm Equivalent (Leica Optics)
Recording Format	4K 29.97 H.264	Full HD 30fps

Table 8: Comparison of XT-5 and Huawei P30 Pro Recording Settings Construction Site A-Bau - Scene: `side_wing`

The subsequent chapter will present the results of the image recordings from the Industrial Solutions Laboratory and Construction Site A-Bau, along with the methods employed, including structured and unstructured approaches, fixed recording heights, and circular recording.

5 Nerfstudio Dataset Pre-Processing

5.1 Accurate Camera Pose Estimation from Images Using COLMAP

COLMAP, a universal SfM and Multi-View Stereo (MVS) pipeline, is used in Nerfstudio to convert self-captured datasets into the required format. It offers a wide range of functions for reconstructing ordered and unordered image collections [Tut23]. The process begins with triangulation, where multiple views of the same object feature are used to calculate the X,Y and Z coordinates of that point in space.

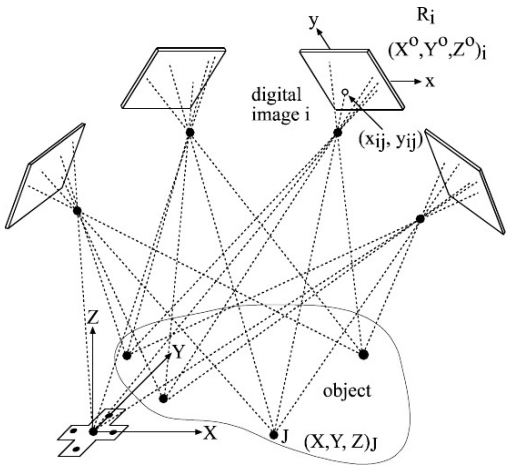


Figure 14: Triangulating XYZ coordinates using multiple views of an object.
Image source: [RF02]

To train models in Nerfstudio on self-captured data sets, the data must be converted into the Nerfstudio format. In particular, the camera positions must be identified for each image. In order to calculate the viewing direction for each image, which is a fundamental input for NeRF, we employ the technique of SfM. The utilization of these algorithms serves to facilitate the process of camera calibration, which is of particular significance for the acquisition of both intrinsic and extrinsic camera parameters. The intrinsic parameters, which include the focal length, principal point coordinates, and lens distortion coefficients, provide insight into the internal configuration of the camera. In contrast, the extrinsic parameters, which encompass rotation and translation matrices for each image, specify the camera's relative orientation and position within a scene [BK11].

For COLMAP (and similar SfM software) to function effectively, it is essential that the intrinsic parameters do not change fundamentally over time for several reasons:

- **Consistency in Calibration:** Intrinsic parameters, such as focal length, principal point, and lens distortion coefficients, define how the camera projects 3D points into the 2D image plane. If these parameters change, the relationship between the 3D world and the 2D images becomes inconsistent, making it difficult for COLMAP to correctly match features across images and accurately reconstruct the scene [PKG99].
- **Feature Matching:** SfM relies on detecting and matching features across multiple images. Consistent intrinsic parameters ensure that the appearance of features remains stable, which improves the robustness and accuracy of feature matching. Changes in intrinsics can cause apparent distortions or shifts in the image, complicating the feature matching process [Low04].
- **Accurate Camera Pose Estimation:** Estimating the camera's position and orientation (extrinsic parameters) requires a stable internal camera model. Variations in intrinsics can lead to errors in pose estimation, as the software might incorrectly interpret the image data, leading to inaccuracies in the reconstructed 3D model [HZ04].
- **Overlap:** In order to carry out the COLMAP processing, it is crucial to identify the same features of an object in multiple images. This necessitates a certain degree of overlap between the images. For stable post-processing, the overlap should be around 80 % within the recording of a scene and 60 % between related recording paths. These values not only ensure the stability of the post-processing but also provide a margin of safety to account for potential errors, such as the Recipient moving faster than expected due to surrounding influences [WW23].

The COLMAP pipeline starts with the acquisition of images from different perspectives of a scene. These images undergo a 'correspondence search,' where features are extracted and matched for geometric consistency. This can include corners, Scale-Invariant Feature Transform (SIFT) points, or other interest points [Low04]. The pipeline then progresses to 'incremental reconstruction,' which includes initialization, image registration, triangulation, and bundle adjustment to optimize the 3D structure [SF16].

To provide a comprehensive overview of the SfM process triggered by the `ns-process` command, we have illustrated the pipeline in Figure 15.

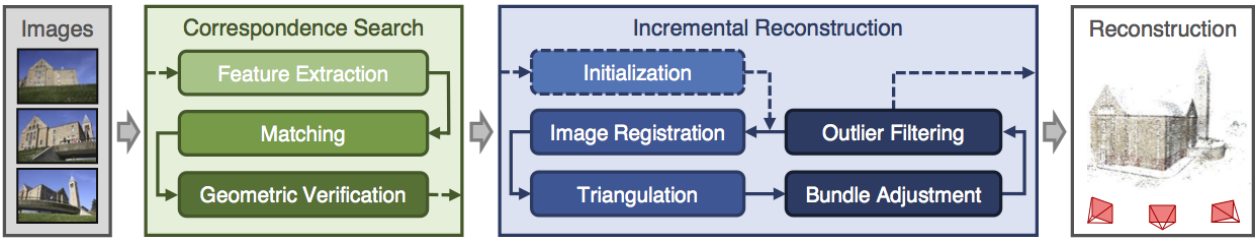


Figure 15: The COLMAP pipeline starts with a set of images and proceeds through various stages including correspondence search, feature extraction, matching, and geometric verification. It then moves on to incremental reconstruction, encompassing initialization, image registration, triangulation, bundle adjustment, and outlier filtering.

Image source: [Tut23]

5.2 Comparative Analysis of Diverse Input Data

5.2.1 Industrial Solutions Laboratory

The objective of the recording was to document the robotic claw arm in the data records as `robot_blue`. By doing so, it was possible to simulate a potential recording for maintenance purposes or for the documentation of installed tools. For this purpose, the results of the untrained user (a) and the trained user (b) utilizing the HUAWEI P30 Pro for pre-processing with COLMAP will be compared in the following.

The following command was used to train the data (`robot_blue`):

```
ns-process-data {video,images,polycam,record3d} --num-frames-target 100 --data
    {DATA_PATH} --output-dir {PROCESSED_DATA_DIR}
```

Dataset	robot_blue_a
Duration:	19 seconds
Number of Matched Images:	118/118
pre-processing Method:	Images were preprocessed using COLMAP.
Clean Up Data:	53/118 Images (45% sharp)
Results:	The outputs presented in Figure 16 and 17
Process Data Command:	`ns-process-data video --num-frames-target 100 --data input/robot_blau_a/VID_20240629_151651_A.mp4 --output-dir processed/robot_blau_a`
Train Model Command:	`ns-train nerfacto --data processed/robot_blau_a`

Dataset	robot_blue_b
Duration:	41 seconds
Number of Matched Images:	104/104
pre-processing Method:	Images were preprocessed using COLMAP.
Clean Up Data:	76/104 Images (73% sharp)
Results:	The outputs presented in Figure 16 and 17
Process Data Command:	`ns-process-data video --num-frames-target 100 --data input/robot_blau_b/VID_20240629_151739_B.mp4 --output-dir processed/robot_blau_b`
Train Model Command:	`ns-train nerfacto --data processed/robot_blau_b`

Table 9: Input data processing and training with Nerfacto in detail for `robot_blau_a` and `robot_blau_b`

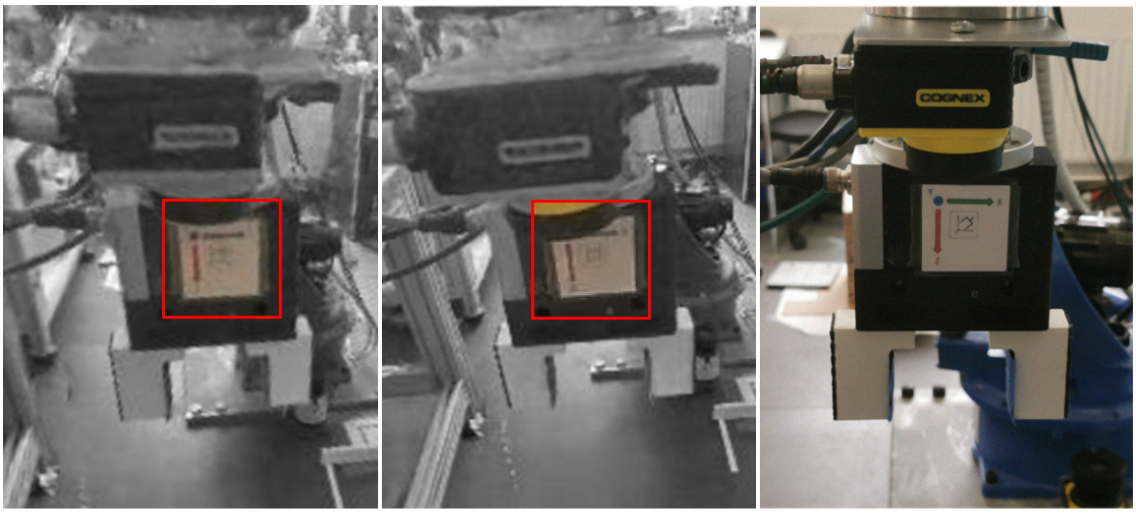
Despite the larger amount of input data for the dataset `robot_blue_a`, the resulting visual quality is inferior to that of `robot_blue_b`. The dataset `robot_blue_a` exhibited a higher prevalence of artifacts throughout the entire scene. Additionally, the highlighted point of interest in `robot_blue_a` was rendered with less visual detail compared to `robot_blue_b`.

In `robot_blue_a`, the automatic settings were utilized for the data recording. Conversely, in `robot_blue_b`, these factors were taken into account. The resulting differences are attributed to the data entered. The following settings were employed in robot_blue_b: camera mode (manual), focus (MF), shutter speed (medium), ISO (high), metering mode (matrix), stabilization (external). It is important to note that during data processing, we identified an error in the camera settings. Instead of recording at 4K resolution with 30 fps, we mistakenly recorded at Full HD with 30 fps. Nevertheless, clear differences in the output quality are observable between the two datasets. In the following section, we proceeded to verify and refine the image frames that had been randomly selected by the process method. To achieve this, we proceeded to remove all images from the training data set for the Nerfacto training that were deemed to be sufficiently recognizable as blurred, and then initiated the training process anew.

The implementation of these steps led to a significant and observable improvement, which can be viewed in the illustrations 16 and 17 in the result of the Nerfacto method.

a) unstructured record b) structured record

Figure 16: We utilize the Nerfstudio viewer for a comparison of visual quality between two datasets captured with different camera settings. a) Unstructured Record (robot_blue_a): This dataset was captured using automatic camera settings, resulting in a higher prevalence of artifacts and less visual detail in the highlighted point of interest. b) Structured Record (robot_blue_b): This dataset was captured with manual camera settings, including manual focus, medium shutter speed, high ISO, matrix metering mode, and external stabilization. The result shows fewer artifacts and greater visual detail in the highlighted area, demonstrating the impact of controlled capture settings on the output quality.
Image source: [Wid24]



a) unstructured cleaned record b) structured cleaned record c) ground truth

Figure 17: We utilize the Nerfstudio viewer for comparative analysis of visual output quality after refining image frames through the removal of blurred data. a) Unstructured Cleaned Record: This image represents the output from the unstructured dataset (robot_blue_a) after removing frames identified as blurred. Despite the cleaning process, the output still shows noticeable artifacts and reduced clarity compared to the structured dataset. b) Structured Cleaned Record: This image represents the output from the structured dataset (robot_blue_b) after similar cleaning procedures. The result demonstrates superior clarity and reduced artifacts, indicating the effectiveness of manual settings and the cleaning process in improving visual quality. c) Ground Truth: This image provides the reference for comparison, showing the actual appearance of the object of interest. It serves as a benchmark to assess the fidelity of the outputs from both datasets.
Image source: [Wid24]

5.2.2 Construction Site Furtwangen University

The objective of this recording was to document the construction process of the elevator, identified in the datasets as `elevator_unboarded`. The recording was intentionally created to generate a dataset that could be used for potential construction process documentation, maintenance purposes, or the documentation of installed cable runs, among other uses. To achieve this, the results of recording Method A and recording Method B will be compared during pre-processing using COLMAP. The recording was conducted using the Fujifilm XT-5 camera and the SAMYANG 12mm 1:2.0 NCS CS X lens.

The following command was used to train the data (`elevator_unboarded`):

```
ns-process-data {video,images,polycam,record3d} --num-frames-target 100 --data
    {DATA_PATH} --output-dir {PROCESSED_DATA_DIR}
```

Dataset	elevator_unboarded_A
Duration:	259 seconds
Number of Matched Images:	36/101
pre-processing Method:	Images were preprocessed using COLMAP.
Clean Up Data:	99/101 Images (98% sharp)
Results:	The outputs presented in Figure 16 and 17
Process Data Command:	`ns-process-data video --num-frames-target 100 --data input/elevator_unboarded_a/DSCF2917_A.mov --output-dir processed/elevator_unboarded_A`
Train Model Command:	`ns-train nerfacto --data processed/elevator_unboarded_A`

Dataset	elevator_unboarded_B
Duration:	115 seconds
Number of Matched Images:	104/104
pre-processing Method:	Images were preprocessed using COLMAP.
Clean Up Data:	100/103 Images (97% sharp)
Results:	The outputs presented in Figure 16 and 17
Process Data Command:	`ns-process-data video --num-frames-target 100 --data input/elevator_unboarded_B/DSCF2918_B.mov --output-dir processed/elevator_unboarded_B`
Train Model Command:	`ns-train nerfacto --data processed/elevator_unboarded_B`

Table 10: Input data processing and training with Nerfacto in detail for `elevator_unboarded_A` and `elevator_unboarded_B`

The dataset `elevator_unboarded_A` presents an anomalous characteristic, specifically an unusually extended video file duration. This extension is directly attributable to the difficulties encountered during the intended utilization of the gimbal used. The inherent challenge in maintaining both the stabilization of the gimbal and the simultaneous focus on the target object necessitated numerous adjustments throughout the recording process. Consequently, this led to an unintended prolongation of the recording time.

To preserve a consistent standard across the utilized datasets, it was necessary to select a fixed number of frames from a substantially larger set of captured frames in the dataset `elevator_unboarded_A`. This selection process, however, diverges from the principle of overlapping content, which typically ensures a higher degree of redundancy and relevance between selected frames. As a result, the dataset exhibits a reduced amount of overlapping content between frames. Due to the limited number of specific camera positions, the NeRF model had to be trained with a significantly reduced amount of input data. This limitation is clearly illustrated in Figure 18. The sparse distribution of camera angles constrained the diversity of the perspectives available for training, thereby impacting the model's ability to fully reconstruct the scene with high fidelity. The implications of this limitation are crucial to understanding the performance limits of NeRF, if unclean or missing input data is used for training, NeRF is only able to compensate for these errors up to a certain limit.

Nevertheless, the methodology employed indicates that, when appropriate camera parameters are meticulously considered, the extraction of a sufficient number of sharp frames can still be achieved. This finding underscores the importance of parameter optimization in the context of video frame extraction, particularly when dealing with extended and complex recordings.

a) structured record, recording technique A, orbital recording, fixed heights

b) structured record, recording technique A, orbital recording, fixed heights

Figure 18: We utilize the Nerfstudio viewer to illustration of the effects if not enough camera positions were identified and the NeRF is trained accordingly with too sparse input data. a) Shows that even if at first glance a sufficient reconstruction of the scene seems to have taken place. b) Illustrates very well that only one side of the scene has sufficient, matched, and corresponding camera positions were transmitted as input data of corresponding quality.

Image source: [Wid24]

The dataset `elevator_unboarded_B` benefited from a freer range of movement during recording, which significantly reduced the required recording time. However, it should be noted that this efficiency could be further improved. Due to suboptimal lighting conditions, the recording had to be conducted at a relatively slow pace, which limited the potential time savings.

Nevertheless, `elevator_unboarded_B` clearly demonstrates that a well-planned and systematically executed recording process substantially enhances the quality of the captured data. The structured approach allowed for the optimization of camera settings and movement strategies, thereby improving the overall sharpness and relevance of the extracted frames. This finding emphasizes the importance of controlled recording conditions in achieving high-quality datasets, particularly in scenarios involving dynamic movements and challenging environments.

a) structured record, recording technique B, orbital recording, circular motion. b) structured record, recording technique B, orbital recording, circular motion.

Figure 19: We utilize the Nerfstudio viewer to visualize how the dataset `elevator_unboarded_B` benefited from a freer range of movement during recording, as demonstrated by the two views presented. a) Highlights the frontal perspective of the recorded structure. b) The lateral perspective is of particular importance in this context, as it allows for a more nuanced understanding of the spatial context and the surrounding environment. In comparison to the previously discussed `elevator_unboarded_A` 18, it is evident that the side view of the elevator can also be reconstructed with greater accuracy, thanks to the larger and more comprehensive data set.

Image source: [WID24]

5.2.3 Practical Guidelines for Dataset Acquisition

To achieve high-quality 3D reconstructions, careful attention must be paid to the dataset acquisition process. The following guidelines outline key practices for ensuring that your images and videos are well-suited for Structure-from-Motion (SfM) processing.

Image Quality Assessment

- After capturing images or videos, thoroughly review each element by zooming in to assess the level of detail. To facilitate this, we always had a laptop on hand to check the recordings in real-time.
- Consider the following questions:
  - Does each image contain sufficient detail to accurately reconstruct the scene?
  - Is the level of detail in each image aligned with the desired outcome for the intended application?
- Eliminate any images that are out of focus, blurry, or otherwise unsuitable. Removing poor-quality images before processing can significantly enhance the overall quality of the results—no information is better than bad information.

Prioritize High-Confidence Information

- Provide the software only with high-quality, high-confidence images to ensure the best possible output.
- Ensure that the background is sharp to capture as many feature points as possible.
- Discard any images that fail to align properly. It is more effective to filter out problematic images before processing than to rely on the software to manage them.

Object Consistency

- Ensure that objects remain consistent throughout the shooting process to avoid discrepancies in the reconstruction.
- Structure-from-Motion (SfM) relies on the assumption that all images depict the object in the same state from different angles, as if captured simultaneously.
- Avoid changes in lighting, shadows, or other factors that could alter the appearance of the object between shots, as these can lead to inconsistencies in the final model.

Ensure Sufficient Keypoints

- Images should contain a high number of keypoints, which are corner points where contrast changes significantly.
- Look for areas with:
  - Varied textures and patterns
  - Objects with distinct edges
  - Text and patterns on surfaces like carpet, newspapers, etc.
- Avoid capturing objects with large regions of monotonous texture or color, as this can lead to poor alignment due to a lack of distinguishable keypoints.

Optimize Shooting Trajectory

- Use a circular shooting trajectory with a constant radius for optimal SfM processing.
- Avoid camera tilts exceeding 30° to minimize perspective distortions.
- This approach ensures the object maintains a consistent size within each image, maximizing detail capture and improving alignment quality.
- Avoid overly complex shooting trajectories, as varying distances from the object can reduce the amount of detail captured, negatively impacting the reconstruction process.

Figure 20: Overview of the utilized hardware: A laptop is crucial for detailed inspection of the recorded data. However, it's equally important to consider the environment where the data is captured. Technical equipment should never be placed in construction dust.
Image source: [Wid24]

Adherence to these guidelines will markedly enhance the precision and caliber of 3D reconstructions when employing SfM technology for data pre-processing, culminating in the generation of more dependable and comprehensive virtual representations.

6 Neural Radiance Fields as an Immersive Technology for Industrial Applications

6.1 Digital Representation of a Construction Site

In the industry, virtual spaces are increasingly utilized for a variety of purposes, including worker training, process simulations, and quality control. However, a significant challenge with these spaces is that, while visually engaging, they often resemble video games more than real-life scenarios. This lack of realism can hinder their effectiveness in practical applications. To address this issue, we present an approach using NeRF that enhances the photorealism of virtual spaces, making them more suitable for industrial use. The recording was conducted using the Fujifilm XT-5 camera and the SAMYANG 12mm 1:2.0 NCS CS X lens.

We successfully demonstrated that reconstructions trained with adequately captured datasets can achieve photorealistic quality in terms of completeness and level of detail. This is achieved by following the methodologies discussed in Chapter 4, *"Smartphone-Based Dataset Acquisition"* and Chapter 5, *"Nerfstudio Dataset Pre-Processing"*. However, for users, it is not only the visual quality that matters but also the range of applications this technology can offer. One potential application is representing these captured spaces in Virtual Reality (VR). While we have not yet explored VR integration, we decided to focus on another practical application: enhancing user interaction with the scene using a web-based viewer with a more user-friendly input device.

6.1.1 Interaction with the Scene Using Nerfstudio Viewer

Using Nerfstudio's web-based viewer, users can observe the training process in real-time. The viewer is automatically launched with each `ns-train` command but can also be started separately with the `ns-viewer` command. To access the viewer, you need to enter the server's address and port into a web browser. However, we advise against monitoring the training progress live through the viewer, as this can significantly slow down the training speed. Once training is complete, the viewer provides a user-friendly interface to inspect the trained scene. The viewport displays all objects in the scene, including the camera positions used for training, which can be toggled on or off.

Navigation and Control Navigation within the viewer is intuitive:

- **Mouse**: Move the scene.
- **Keyboard**: Use the `W, A, S, D, Q, E` keys to move around the scene and the `arrow` keys to rotate.

The control panel allows various training-related settings to be adjusted, such as changing the training speed or visualizing different output modes. In the scene panel, users can control the visibility of individual objects by clicking on the eye icon next to them. This feature is particularly useful for focusing on specific aspects of the visualization by hiding irrelevant elements.

For creating a custom camera path, we refer to Chapter 3.3.3, *"Creating and Exporting the Flythrough"*. While creating, rendering, and exporting a scene flythrough with Nerfstudio is intuitive, the interaction and potential of a continuously displayed scene are limited by this approach. Therefore, we took the concept of a scene flythrough literally and explored the possibilities of immersive interaction within the built-in web viewer.

6.1.2 Enhancing Navigation with a Controller

The motivation behind enhancing navigation stems from the need to move freely within the 3D scene. Since the laptop used had no connected mouse, navigation within the viewer was somewhat cumbersome. Therefore, we decided to map this navigation to a controller, offering several advantages:

1. **Ergonomics and Comfort**: The controller is ergonomically designed, fitting comfortably in the hand even during extended use. Simultaneously using `WASD`, `arrow keys`, and a `mouse` requires constant hand repositioning, which is unnecessary with the controller. The controller's buttons and joysticks allow for simultaneous operation with both hands, enabling users to navigate through the scene without taking their eyes off it.
2. **Analog Control**: The joysticks on the controller provide analog control, allowing for more precise movements and finer control. This is particularly important when examining smaller details, where precise camera movements should be ensured.
3. **Freedom of Movement**: Controllers allow for more intuitive and straightforward freedom of movement in 3D environments. The simultaneous control of both the viewing direction and movement is smoother compared to using a mouse and keyboard.
4. **Feedback**: Haptic feedback (vibration) could be used to convey collisions within the scene.

Since Nerfstudio does not have built-in controller support, we used the `Joy2Key` software to map the controls. `Joy2Key` is a software tool that allows controllers, like the Microsoft Xbox controller, to emulate keyboard and mouse input, thereby enabling control of Windows applications and web-based applications using the controller as an input device [Joy].

6.1.3 Case Study: Virtual Walkthrough of a Construction Site

To evaluate the effectiveness of our approach, we conducted a virtual walkthrough of a construction site, in illustration 21, focusing on inspecting a laboratory exhaust system. This simulated a real-life site visit to assess the level of detail and coverage within the scene. The dataset used for this purpose was `air_system`. To improve movement fluidity, we adjusted the resolution in the Nerfstudio web viewer to address frame rate limitations imposed by the hardware. The point of view (PoV) was controlled using the left and right joysticks of a Microsoft Xbox controller. The results, influenced by hardware limitations, will be evaluated separately for maneuverability and optical outcomes.

a) visual quality while navigating through the scene

b) optical quality while no movement was performed at high render quality

Figure 21: a) Despite the noticeably adjusted image quality of the viewer, we were able to easily reach every point in the room using the controls. Although it could be expected that certain areas might not be targeted effectively due to their lack of sharpness, we found it straightforward in the context of the scene, such as the "fuse box". b) This illustration highlights the potential improvements achievable with more powerful hardware. While individual details remain somewhat indistinct due to the set rendering quality, the overall display of the scene has significantly improved, demonstrating the benefits of enhanced hardware capabilities. Image source: [WID24]

6.1.4 Conclusion

In this scenario, we demonstrated that video or image data from a construction site can be effectively utilized to virtually inspect the site. The use of a customized control system further enabled an intuitive examination of the scene. The entire room was fully captured within approximately 5 minutes, with subsequent data processing and NeRF training taking an additional 35 minutes. This efficiency suggests that rooms, complexes, or even entire buildings can be quickly captured and navigated in digital form. By integrating multi-camera setups or using more powerful hardware, current limitations in live rendering could be addressed and potentially overcome.

This approach demonstrates significant potential for efficient and accurate virtual reconstructions, allowing for quick and immersive inspections of physical spaces. Future work could explore the integration of VR for an even more immersive experience or enhancements in photorealism through advanced NeRF techniques.

6.2 Applying Text-based Search for Objects Utilizing LERF

LERF represents a new advancement in NeRF technology, enabling computers to identify specific items within a NeRF scene based on textual descriptions. This innovation unlocks a wide range of applications, including assembly instructions, building inspections, and maintenance work. With LERF, it becomes possible to immediately locate the content you're searching for by entering a text query, rather than manually sifting through a video or image dataset.

6.2.1 Challenges in Traditional Image Search

Consider a scenario where you are recording a section of a production hall for structural documentation. The goal is to identify all instances of damage to detect potential structural defects at an early stage. After recording, you have a dataset of 731 images. As you review the images, you keep a tally of the damage found.

After about 21 images, you notice a feature in one image that is no longer visible in the next due to the angle of capture. You could manually search for the matching image that clearly shows this feature, akin to piecing together a jigsaw puzzle. While this approach may work for a few features, it quickly becomes impractical as the number of features to search for increases. On the other hand, if you're only looking for a few specific features within the 731 images, you're faced with a different challenge: finding a needle in a haystack. This is where LERF offers a transformative and highly efficient solution.

6.2.2 Capabilities of LERF

LERF enables pixel-aligned queries of the distilled 3D CLIP embeddings without relying on region proposals, masks, or fine-tuning, supporting long-tail, open-vocabulary queries hierarchically across the volume [Ker+23]. The level of detail that LERF can achieve is remarkable. It not only understands larger scene contexts but also isolates individual elements within them.

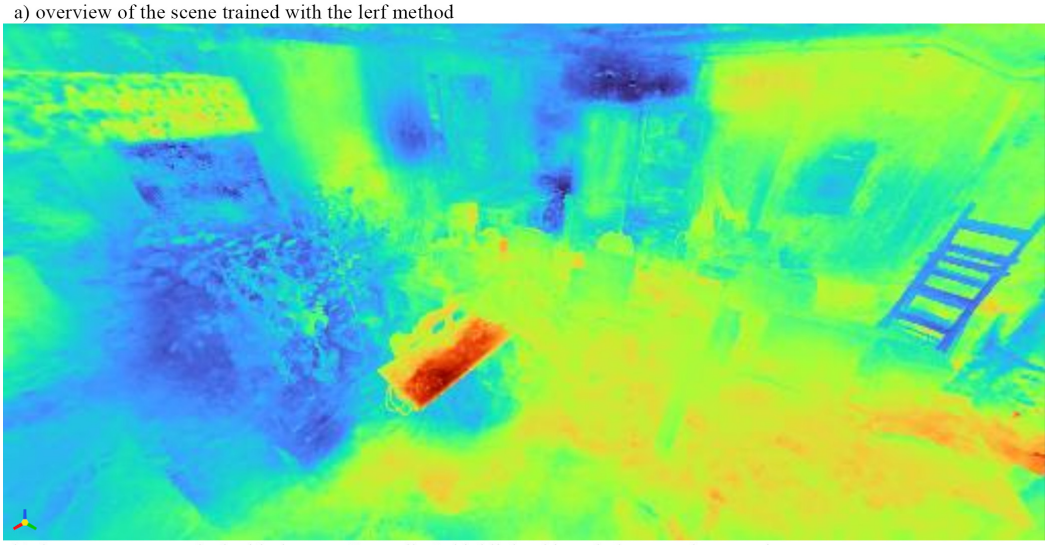
6.2.3 Practical Example: Searching for Objects with LERF

We have illustrated this with a practical example in Figure 22. The recording was conducted using a Fujifilm XT-5 camera and a SAMYANG 12mm 1:2.0 NCS CS X lens, capturing a section of the Industrial Solutions Lab in 261 images. In a self-experiment, we tested how long it took four different test subjects to find three specific objects within these images: a ladder, a controller, and a cube. While our test was limited to the number of captured images in the scene `work_place_01`, it is evident how much more complex it would be to search the entire laboratory for the specified content. Unfortunately, we were unable to practically demonstrate this example due to insufficient system performance, which necessitated a reduction in the scale of the application scenario.

We deliberately started with a relatively large object to ensure at least one success. However, it quickly became apparent how challenging it is to search for objects in a scene when you don't know exactly what they look like or where they are. While all four test subjects found the ladder and the controller, and after an average of 4 minutes, also found a cube, none of them discovered the second cube or the smaller cubes hidden in the scene. All the test subjects assumed they had completed the task once they found the three objects they were instructed to locate.

This simple example highlights why methods like LERF are invaluable for searching through complex scenes. LERF provides a systematic and efficient way to quickly and reliably identify specific features in large image datasets.

a) overview of the scene trained with the lerf method



b) the scene was searched with the term "conroller", highlighted in red, the area closest to it

Figure 22: a) PoV of a scene trained in Nerfstudio using the LERF method. Notice how the perspective is significantly elevated above the captured scene, illustrating how users can break free from existing viewpoints and explore scenes from new angles. b) The same scene is depicted again, but this time with the object search for "Controller" applied. The area highlighted in red marks the region identified by LERF as the content area with a match for "Controller."

Image source: [Wid24]

6.2.4 Conclusion of Text-based Search for Objects Utilizing LERF

The integration of Language Embeddings into NeRF introduces a wide array of potential applications. The LERF method facilitates interaction with a scene through text input, functioning similarly to a search engine but within a three-dimensional scene. This allows for the rapid and accurate identification of objects and scene relationships, significantly enhancing the efficiency and precision with which points of interest can be located.

In our practical example, participants took an average of 4 minutes to locate an unspecified number of objects within the scene. However, the experimental setup revealed that more objects were detectable than what was originally defined as a single object per search. This discrepancy underscores a crucial advantage of the LERF approach, particularly in real-world applications where the ability to quickly and accurately identify multiple objects within a complex scene can be critically important.

Moreover, the ability to search and visualize 3D spaces using natural language input opens new possibilities for fields such as virtual reality, robotics, and digital content creation. The efficiency gains observed in our study suggest that LERF could greatly enhance user interaction in these domains, enabling faster decision-making and reducing the cognitive load associated with navigating and interpreting complex visual environments.

7 Conclusions

7.1 Summary

In this study, we conducted a comprehensive examination and explored the potential NeRF applications in the industrial sector. Our investigation has unveiled that NeRFs can overcome limitations associated with traditional 3D representations and rendering methods, such as their high cost, equipment requirements, and limited realism. Furthermore, we investigated innovative NeRF applications in industrial settings and substantiated their viability through proof-of-concept experiments. Our goal was to provide a comprehensive guide tailored to users without a technical background, enabling them to understand the essential processes of data capture and processing within real-world application scenarios. The overarching theme of this work is the practical application of NeRF technology. Therefor we outlined the technological evolution and the growing significance of 3D visualization techniques, as discussed in Chapter 1. This set the stage for a deeper exploration of the NeRF algorithm, which has garnered significant attention in the field of computer vision for its ability to accurately capture and recreate real-world scenes, including intricate lighting properties based on viewing angles. The potential of this algorithm is particularly notable in applications, where photorealistic reconstruction are crucial.

In Chapter 2, we delved into the theoretical foundations of NeRFs, providing an overview of the algorithm's capabilities and the enhancements that have been made since its introduction. We highlighted how NeRF improvements, such as NeRF-W and Nerfacto, have expanded its applicability to more complex, real-world scenarios. This discussion provided the necessary context for understanding the specific contributions of the Nerfstudio framework, which were further elaborated in the subsequent chapters. Chapter 3 served as an introduction to the Nerfstudio framework, offering users an initial hands-on experience with the software interface. We demonstrated the workflow for processing and training NeRF models, including detailed instructions for installing the software using Docker, which addressed common challenges encountered during setup. This chapter was designed to familiarize users with the tools and techniques required for effective NeRF implementation.

In Chapter 4, we shifted our focus to data acquisition using consumer-grade SPC electronics, such as the HUAWEI P30 Pro, with an emphasis on reproducibility and the impact of various factors on the quality of datasets. We provided a thorough analysis of smartphone camera functionalities, laying the groundwork for understanding how these devices can be used effectively in NeRF dataset capturing. This chapter also explored the importance of optimizing camera settings to ensure consistent, high-quality data capture, which is vital for the success of subsequent 3D reconstructions. Chapter 5 delved into the pre-processing of datasets within the Nerfstudio framework. We discussed the importance of accurate camera pose estimation using tools like COLMAP and conducted a comparative analysis of input data from different environments.

This chapter provided practical guidelines for dataset preparation, emphasizing the importance of meticulous pre-processing to achieve optimal results in NeRF training.

In Chapter 6, we explored the application of NeRF technology in industrial contexts, using case studies to illustrate its potential. We demonstrated how NeRF can be used to create immersive, interactive 3D representations of industrial environments, such as construction sites. This chapter highlighted the transformative impact of NeRF on industrial applications, showcasing its ability to enhance digital representations and facilitate better interaction with 3D scenes. Our findings indicate that a comprehensive understanding of the capabilities and limitations of NeRFs is essential for fully leveraging the algorithm's potential. Our results provide a foundation for future research endeavors in this dynamic and evolving field.

7.2 Outlook

A growing number of companies are developing applications based on NeRF technology, driven by increasingly sophisticated models and frameworks that enable a broader range of applications. The widespread adoption and societal impact of LLMs underscore the versatility of AI, not only in computer graphics but across various fields.

Moreover, the importance and diversity of multi-agent systems continue to expand, encompassing robotics, web agents, financial systems, video games, sensor networks, and any agents that interact with humans. In the future, autonomous agents will increasingly assist us in our daily lives. Advances in robotics, which could perhaps be trained in a virtual NeRF environment hint at the vast potential of future applications for multi-agent systems. NeRF technology is poised to play a significant role in this evolution, potentially expanding its applications in ways previously unimaginable. However, it remains to be seen how research and development will meet market expectations and whether efficient, practical solutions for integrating these technologies can be achieved.

In conclusion, this thesis highlights the emergence of a new and promising field of research driven by NeRF technology. The groundwork established by NeRF offers a robust foundation for future research to explore diverse and innovative directions. This work not only emphasizes the significance of NeRF as a pioneering technology for industrial applications but also underscores the vast potential it unlocks for further studies and applications.

8 Attatchments

8.1 Images

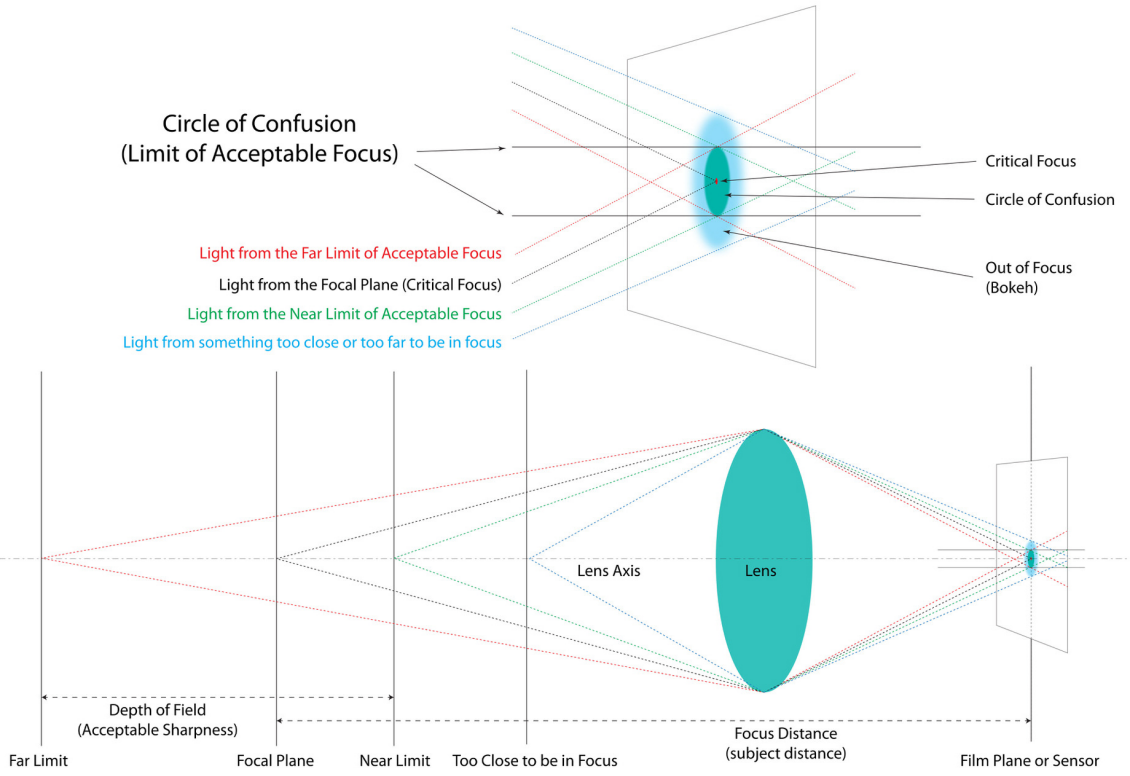


Figure 23: Circle of Confusion Diagram [Bai21].
Image source: [Bai21]

8.2 Instructions for Dockerfile Adjustments

Please make the following adjustments to the Dockerfile:

1. **Line 76 and 131:**
   - Add the package "`python-setuptools \`" to the existing list of packages to be installed, ensuring that setuptools is correctly installed.
   - Under # Upgrade pip and install packages, add "`RUN python3.10 -m pip install --no-cache-dir --upgrade pip setuptools==69.5.1 pathtools promise pybind11 omegaconf`" to the existing list of packages.
2. **Package Installation:**
   - Include the installation of the package `numpy==1.26.4`. This can either be integrated into the current package installation list or executed as a separate command.

9 Utilization of Artificial Intelligence Technologies

9.1 DeepL Write

DeepL Write (HTTPS://WWW.DEEPL.COM/WRITE) is an advanced text-editing service that leverages machine learning technologies. The tool is specifically designed to augment the quality of text through spelling, grammar, and sentence structure corrections.

In this research, DeepL Write was utilized for multiple objectives:

- Text Correction: The tool played a significant role in enhancing spelling, grammar, and sentence structure.
- Quality Enhancement: DeepL Write contributed to elevating the overall textual quality of the thesis to an academically rigorous standard.

9.2 Chat-GPT

Chat-GPT (HTTPS://CHAT.OPENAI.COM/), also referred to as the Generative Pre-trained Transformer (GPT), is a sophisticated conversational agent developed by OpenAI. Based on the Transformer architecture, it has been trained on an extensive dataset encompassing a wide range of content from the internet. The model is highly proficient in generating text that closely resembles human language and has a broad spectrum of applications, from automating customer service to supporting academic research.

In this thesis, Chat-GPT was deployed for multiple purposes:

- Caption Annotation: Chat-GPT assisted in the labeling and descriptions of captions.
- Content Review: Various sections of this research were reviewed and enhanced through Chat-GPT's generative capabilities.
- Formulation Assistance: In instances where phrasing or formulation posed challenges, Chat-GPT was employed to provide alternative expressions or clarify complex ideas.

10 Declaration on Honest Academic Work

With this document, I, Dominik Widmann, declare that I have drafted and created the piece of work in hand myself. I declare that I have only used such aids as are permissible and used no other sources or aids than the ones declared. I furthermore assert that any passages used, whether verbatim or paraphrased, have been cited in accordance with current academic citation rules and such passages have been marked accordingly. Additionally, I declare that I have disclosed and stated all uses of any aids such as AI-based chatbots (e.g., ChatGPT), translation tools (e.g., DeepL), paraphrasing tools (e.g., Quillbot), or programming devices (e.g., GitHub Copilot) and have marked any relevant passages accordingly.

I am aware that the use of machine-generated texts does not guarantee the quality of their content or the text as a whole. I assert that I used text-generating AI tools merely as an aid and that the piece of work in hand is, for the most part, the result of my creative input. I am entirely responsible for the use of any machine-generated passages of text I have used.

I also confirm that I have taken note of the document "Satzung der Hochschule Furtwangen (HFU) zur Sicherung guter wissenschaftlicher Praxis" dated October 27, 2022, and that I have followed the statements therein.

I am aware that my work may be examined to determine whether any non-permissible aids or plagiarism were used. I also acknowledge that a breach of § 10 or § 11 sections 4 and 5 of HFU's study and examination regulations' general part may lead to a grade of 5 or "nicht ausreichend" (not sufficient) for the work in question and/or exclusion from any further examinations.

Place, Date, Signature

Dominik Widmann

Furtwangen, 30.08.2024

11 Figures

List of Figures

References

[24a] *Hyperfocal distance*. [Accessed: Jun. 18, 2024]. 2024. URL: https://www.dofmaster.com/hyperfocal.html.

[24b] *Image Requirements for Photogrammetry – Strucinspect*. [Accessed: Jul. 26, 2024]. 2024. URL: https://academy.strucinspect.com/lessons/2-image-requirements-for-photogrammetry/.

[24c] *ImageHistogram — Scientific Volume Imaging*. [Accessed: Jul. 22, 2024]. 2024. URL: https://svi.nl/ImageHistogram.

[24d] *Orthomosaic*. [Accessed: Jun. 04, 2024]. 2024. URL: https://support.sitemark.com/en/articles/4592264-orthomosaics-aesthetic-vs-functional.

[24e] *Understanding Digital Camera Histograms: Tones and Contrast*. [Accessed: Mai. 28, 2024]. 2024. URL: https://www.cambridgeincolour.com/tutorials/histograms1.htm.

[Ado] Adobe. *What is 3D Modeling & What is it Used For?* https://www.adobe.com/products/substance3d/discover/what-is-3d-modeling.html. [Accessed: Aug. 12, 2024].

[Ame21] American Psychological Association. *Learning and Memory*. https://www.apa.org/topics/learning-memory. [Accessed: Mai. 17, 2024]. 2021.

[Art] Artec 3D. *What is photogrammetry*. https://www.artec3d.com/learning-center/what-is-photogrammetry. [Accessed: Aug. 12, 2024].

[Bai21] Martin Bailey. *Depth of Field, Hyperfocal Distance, Infinity, and Beyond! (Podcast 732)*. [Accessed: May. 12, 2024]. Feb. 2021. URL: https://martinbaileyphotography.com/2021/02/18/depth-of-field-hyperfocal-distance-infinity-and-beyond-podcast-732/.

[Bar+21] Jonathan T. Barron et al. *MIP-NERF 360: Unbounded Anti-Aliased Neural Radiance Fields*. Accessed: Aug. 09, 2024. Nov. 2021. URL: https://arxiv.org/abs/2111.12077.

[BK11] Gary R. Bradski and Adrian Kaehler. *Learning OpenCV: Computer Vision with the OpenCV Library*. 1st ed. [Nachdr.] O'Reilly, 2011.

[BM14] Erik Brynjolfsson and Andrew McAfee. *The Second Machine Age: Work, Progress, and Prosperity in a Time of Brilliant Technologies*. New York, NY, USA: W. W. Norton & Company, 2014.

[BS21] Viktor Blahnik and Oliver Schindelbeck. "Smartphone Imaging Technology and its Applications". In: *Advanced Optical Technologies* 10.3 (2021). [Accessed: Jun. 12, 2024], pp. 145–232. URL: https://www.degruyter.com/document/doi/10.1515/aot-2021-0023/html.

[BSI] CMOS vs BSI Sensor. *CMOS vs BSI Sensor*. [Accessed: Aug. 09, 2024]. URL: https://www.nevsemi.com/blog/cmos-vs-bsi-sensor.

[Cam] DSLR Camera. *DSLR Camera: The Ultimate Guide to Getting Started Fast.* [Accessed: Jul. 25, 2024]. URL: https://es.clippingmagic.com/resources/dslr-camera-the-ultimate-guide-to-getting-started-fast.

[DJI] DJI Enterprise. *Ground Sample Distance — DJI Enterprise.* [Accessed: May. 21, 2024]. URL: https://enterprise-insights.dji.com/blog/ground-sample-distance.

[Doc24] Docker Documentation. *Docker overview.* [Accessed: Jul. 21, 2024]. Aug. 2024. URL: https://docs.docker.com/get-started/docker-overview/.

[Edm] Edmund Optics. *Considerations in Collimation.* [Accessed: Jun. 14, 2024]. URL: https://www.edmundoptics.com/knowledge-center/application-notes/optics/considerations-in-collimation/.

[Gee23] GeeksforGeeks. *Nyquist Sampling Rate and Nyquist Interval.* [Accessed: Jul. 27, 2024]. Feb. 2023. URL: https://www.geeksforgeeks.org/nyquist-sampling-rate-and-nyquist-interval/.

[Got97] L. S. Gottfredson. "Mainstream science on intelligence: An editorial with 52 signatories, history and bibliography". In: *Intelligence* 24.1 (1997). [Accessed: Jun. 27, 2024], pp. 13–23. DOI: 10.1016/S0160-2896(97)90011-8.

[HUA23] HUAWEI Support Global. *Pro Mode — HUAWEI Support Global.* [Accessed: Aug. 20, 2024]. 2023. URL: https://consumer.huawei.com/en/support/content/en-us01087371/.

[HZ04] Richard I. Hartley and Andrew Zisserman. *Multiple View Geometry in Computer Vision.* 2nd ed. Cambridge, UK: Cambridge University Press, 2004.

[Jen89] A. R. Jensen. "The relationship between learning and intelligence". In: *Learning and Individual Differences* 1.1 (Jan. 1989), pp. 37–62. DOI: 10.1016/1041-6080(89)90009-5.

[Joy] JoyToKey. *JoyToKey - Download the latest official version.* [Accessed: Aug. 22, 2024]. URL: https://joytokey.net/en/.

[Ker+23] Jack Kerr et al. *LERF: Language embedded Radiance Fields.* [Accessed: Aug. 02, 2024[]. Mar. 2023. URL: https://arxiv.org/abs/2303.09553.

[Lin23] Lu Ling. *: A Large-Scale Scene Dataset for Deep Learning-based 3D Vision.* [Accessed: Aug. 12, 2024]. 2023. URL: https://arxiv.org/html/2312.16256v2.

[LL23] Nadine Lederer and Nick Lederer. *Die richtige Objektivwahl: Von Weitwinkel bis Teleobjektiv.* [Accessed: Aug. 21, 2024]. Dec. 2023. URL: https://www.suitcaseandwanderlust.com/objektivwahl/.

[Low04] David G. Lowe. "Distinctive Image Features from Scale-Invariant Keypoints". In: *International Journal of Computer Vision* 60.2 (2004), pp. 91–110.

[Man+23] T. P. Mantas et al. "Photogrammetry, from the Land to the Sea and Beyond: A Unifying Approach to Study Terrestrial and Marine Environments". In: *Journal of Marine Science and Engineering* 11.4 (Mar. 2023), p. 759. DOI: 10.3390/jmse11040759.

[Mar+20] Ricardo Martin-Brualla et al. *NERF in the Wild: Neural radiance fields for unconstrained photo collections*. [Accessed: Aug. 14, 2024]. Aug. 2020. URL: HTTPS://ARXIV.ORG/ABS/2008.02268.

[Mic24] Microsoft. *Install WSL*. [Accessed: Jul. 05, 2024]. 2024. URL: HTTPS://LEARN.MICROSOFT.COM/DE-DE/WINDOWS/WSL/INSTALL.

[Mil+20] B. Mildenhall et al. *NERF: Representing Scenes as Neural Radiance Fields for View Synthesis*. arXiv.org. [Accessed: Aug. 13, 2024]. Apr. 2020. URL: HTTPS://ARXIV.ORG/ABS/2003.08934.

[Mül+22] Thomas Müller et al. "Instant Neural Graphics Primitives with a Multiresolution Hash Encoding". In: *ACM Transactions on Graphics* 41.4 (July 2022), pp. 1–15. DOI: 10.1145/3528223.3530127.

[NMV22] Enrico Nocerino, Fabio Menna, and Geert J. Verhoeven. "GOOD VIBRATIONS? HOW IMAGE STABILISATION INFLUENCES PHOTOGRAMMETRY". In: *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences* XLVI-2/W1-2022 (Feb. 2022). [Accessed: Jul. 25, 2024], pp. 395–400. DOI: 10.5194/ISPRS-ARCHIVES-XLVI-2-W1-2022-395-2022.

[Ope] OpenCV. *OpenCV: Introduction to SIFT (Scale-Invariant Feature Transform)*. [Accessed: Jul. 24, 2024]. URL: HTTPS://DOCS.OPENCV.ORG/4.X/DA/DF5/TUTORIAL_PY_SIFT_INTRO.HTML.

[Pea19] Ronan Pearson-Wright. *Histogram: Discover How to Take Better Photos by Exposing to the Right*. [Accessed: Mai. 02, 2024]. Jan. 2019. URL: HTTPS://PHOTOGRAPHYPRO.COM/HISTOGRAM/.

[Pix] Pixpro. *Raw Photo Processing for Photogrammetry - Our Easy Workflow*. [Accessed: Jul. 25, 2024]. URL: HTTPS://WWW.PIX-PRO.COM/BLOG/RAW.

[PKG99] Marc Pollefeys, Reinhard Koch, and Luc Van Gool. "Self-Calibration and Metric Reconstruction in Spite of Varying and Unknown Intrinsic Camera Parameters". In: *International Journal of Computer Vision* 32.1 (1999), pp. 7–25.

[Pro24] Proofpoint. *What is a sandbox environment? Meaning & setup*. HTTPS://WWW.PROOFPOINT.COM/US/THREAT-REFERENCE/SANDBOX. [Accessed: Jul. 21, 2024]. Feb. 2024.

[Pum+20] Albert Pumarola et al. *D-NERF: Neural Radiance Fields for dynamic scenes*. [Accessed: Aug. 15, 2024]. Nov. 2020. URL: HTTPS://ARXIV.ORG/ABS/2011.13961.

[Rao23] P. Rao. *How Long it Took for Popular Apps to Reach 100 Million Users*. HTTPS://WWW.VISUALCAPITALIST.COM/THREADS-100-MILLION-USERS/. [Accessed: Jun. 27, 2024]. Oct. 2023.

[RF02] T. G. Ryall and C. S. Fraser. "Determination of Structural Modes of Vibration Using Digital Photogrammetry". In: *Journal of Aircraft* 39.1 (Jan. 2002). [Accessed: May. 28, 2024], pp. 114–119. DOI: 10.2514/2.2903.

[RG16] Evan F. Risko and Sam J. Gilbert. "Cognitive offloading". In: *Trends in Cognitive Sciences* 20.9 (2016), pp. 676–688. DOI: 10.1016/J.TICS.2016.07.002.

[Ros05] Jarek Rossignac. "3D Mesh Compression". In: *Elsevier eBooks*. Elsevier, 2005, pp. 359–379. DOI: 10.1016/B978-012387582-2/50020-4.

[RSK20] A. Rasheed, O. San, and T. Kvamsdal. "Digital twin: Values, challenges and enablers from a modeling perspective". In: *IEEE Access* 8 (2020), pp. 21980–22012. DOI: 10.1109/ACCESS.2020.2970143.

[Rus21] David Russ. *How does the eye work? - Optometrists.org*. [Accessed: Jul. 27, 2024]. Aug. 2021. URL: HTTPS://WWW.OPTOMETRISTS.ORG/GENERAL-PRACTICE-OPTOMETRY/GUIDE-TO-EYE-HEALTH/HOW-DOES-THE-EYE-WORK/.

[SF16] Johannes L. Schönberger and Jan-Michael Frahm. "Structure-from-Motion Revisited". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2016.

[Sta21] Cyrill Stachniss. *Histograms Presentation - Photogrammetry Course*. [Accessed: Aug. 03, 2024]. 2021. URL: HTTPS://WWW.IPB.UNI-BONN.DE/HTML/TEACHING/PHOTO12-2021/2021-PHO1-03-IMG-HISTO-1-HISTOGRAMS.PPTX.PDF.

[Tan+23] Matthew Tancik et al. *Nerfstudio: A Modular Framework for Neural Radiance Field Development*. [Accessed: Jun. 22, 2024]. July 2023. URL: HTTPS://DOI.ORG/10.1145/3588432.3591516.

[Ten23] TencentARC. *GitHub - TencentARC/ArcNerf: Nerf and Extensions in All*. [Accessed: Jul. 12, 2024]. 2023. URL: HTTPS://GITHUB.COM/TENCENTARC/ARCNERF.

[The] The Smartphone Photographer. *How does a smartphone camera work? A detailed walk through*. [Accessed: Jul. 25, 2024]. URL: HTTPS://THESMARTPHONEPHOTOGRAPHER.COM/HOW-PHONE-CAMERA-WORKS/.

[Tut23] Tutorial — COLMAP 3.9. *Tutorial — COLMAP 3.9-dev documentation*. [Accessed: May. 19, 2024]. 2023. URL: HTTPS://COLMAP.GITHUB.IO/TUTORIAL.HTML.

[Ver+21] Dor Verbin et al. *Ref-NERF: Structured View-Dependent Appearance for Neural Radiance Fields*. [Accessed: Jul. 11, 2024]. Dec. 2021. URL: HTTPS://ARXIV.ORG/ABS/2112.03907.

[Wan+21] Zirui Wang et al. *NERF–: Neural Radiance Fields without Known Camera Parameters*. [Accessed: Aug. 03, 2024]. Feb. 2021. URL: HTTPS://ARXIV.ORG/ABS/2102.07064.

[Wid24] Dominik Widmann. *A collection of images showcasing the topics covered*. Unpublished work. 2024.

[WW23] Mark Walker and Mary Walker. *Optimising Overlap in Drone Mapping: A Comprehensive Guide for Operators and Surveyors*. [Accessed: Aug. 16, 2024]. Nov. 2023. URL: HTTPS://WWW.AIRCAMDRONESERVICES.COM/.

[Zha+20] Kai Zhang et al. *NERF++: Analyzing and improving Neural Radiance Fields*. [Accessed: Aug. 12, 2024]. Oct. 2020. URL: HTTPS://ARXIV.ORG/ABS/2010.07492.